UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

TESI DI LAUREA

Progettazione e sviluppo di una Enterprise Big Data Platform ed implementazione di algoritmi di Predictive Maintenance in ambito Automotive

Design and development of an Enterprise Big Data Platform and implementation of Predictive Maintenance algorithms in the Automotive field

Relatore

Prof. Domenico Ursino

Candidata

Silvia Paolucci

Correlatrice

Federica Marini

ANNO ACCADEMICO 2020-2021

Sommario

Oggi tutto ciò che facciamo, in particolar modo a livello informatico, si traduce nella produzione di una serie di dati che navigano la rete e vengono immagazzinati nei server. Basta considerare la quantità di operazioni svolte in ogni istante in tutto il mondo per capire la portata del fenomeno, che prende il nome di Big Data, e giustificare perché esso abbia portato all'immediata necessità di nuove tecnologie e processi in grado di immagazzinare, gestire e sfruttare opportunamente i dati.

La prima parte di questa tesi si occuperà della definizione di un ambiente per l'archiviazione centralizzata dei dati di una casa automobilistica, adatto alla conservazione dei Big Data, ovvero il Data Lake. Entrando, poi, nelle necessità di Business, la seconda parte della tesi presenterà, dapprima, una serie di Use Case di Big Data e *Advanced Analytics*, che sfruttano questi dati per rispondere alle nuove necessità di Business. L'ultima parte della trattazione si concentrerà sullo sviluppo del primo Use Case in ordine di prioritizzazione, ovvero quello di *Predictive Maintenance* su un'auto sportiva, focalizzandosi sulla presentazione del particolare approccio adottato.

Keyword: Big Data, Data Lake; Advanced Analytics; Whole Vehicle Predictive Maintenance; Data Science; Machine Learning; K-means; XGBoost; PySpark

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Oggi tutto ciò che facciamo, in particolar modo a livello informatico, si traduce nella produzione di una serie di dati che navigano la rete e vengono immagazzinati nei server. Basta considerare la quantità di operazioni svolte in ogni istante in tutto il mondo per capire la portata del fenomeno, che prende il nome di Big Data, e giustificare perché esso abbia portato all'immediata necessità di nuove tecnologie e processi in grado di immagazzinarli, gestirli e sfruttarli opportunamente.

Le aziende automobilistiche non sono, di certo, immuni a questa ondata di cambiamento. Anzi, per certi versi, devono anche gestire quantità di dati ancora più elevate. Infatti, in quanto aziende produttive si trovano a dover immagazzinare i dati provenienti dai macchinari, ma anche quelli prodotti a valle delle attività quotidiane dei vari dipartimenti. A questi dati si aggiungono quelli inviati dai veicoli venduti che, grazie alle tecnologie e ai sensori di bordo, fungono da veri e propri dispositivi IoT.

La prima parte di questa tesi si occuperà della definizione di un ambiente di archiviazione centralizzata dei dati per una casa automobilistica, adatto alla conservazione dei Big Data che, per loro natura, provengono da sorgenti varie ed eterogenee, e per i quali non si è in grado di predeterminare uno schema logico.

Entrando, poi, nelle necessità di Business, la seconda parte della tesi presenterà dapprima una serie di Use Case di Big Data e *Advanced Analytics*, che sono stati delineati in risposta alle necessità di Business evidenziate, per poi focalizzarsi soltanto su uno di questi.

L'ultima parte della tesi si concentrerà sullo sviluppo dello Use Case di *Predictive Maintenance* relativo ad un particolare modello d'auto sportiva prodotto dalla casa automobilistica. Verranno fornite informazioni in merito ai dati a disposizione e agli approcci di *Machine Learning* valutati per la definizione del modello predittivo, fino ad arrivare a presentare i risultati ottenuti.

L'elaborato si compone di 8 capitoli, per i quali di seguito verrà fornita una breve presentazione:

- *Capitolo 1* - Fornisce una panoramica dell'azienda automobilistica cliente di Deloitte Consulting. Verrà proposta una *overview* sulla struttura interna, con una breve presentazione delle diverse divisioni che la compongono, per poi focalizzarsi sulla strategia che guida le attività aziendali, nonché sugli obiettivi e sulle necessità di business evidenziate a livello generale.

- *Capitolo 2* - Si entra nel vivo delle necessità di business. Verrà, come prima cosa, discussa e introdotta una tipologia di ambiente di archiviazione dei dati più idonea a rispondere ai nuovi bisogni interni, ovvero il Data Lake, fornendo dettagli sulla sua progettazione

e implementazione. Nella parte conclusiva del capitolo saranno presentati una serie di Use Case, per lo più di Big Data e *Advanced Analytics*, delineati in risposta alle necessità di Business evidenziate. Tanto per l'implementazione del Data Lake quanto per lo sviluppo degli Use Case verranno indicati gli approcci definiti e seguiti da Deloitte.

- *Capitolo 3* - Da qui in poi la trattazione si occuperà unicamente dello Use Case di *Predictive Maintenance* su un particolare modello d'auto sportiva prodotto dalla casa automobilistica. Questo capitolo fornirà dapprima una breve introduzione relativa alla piattaforma cloud che si è deciso di utilizzare per l'implementazione della porzione di Data Lake dedicata, per poi passare alla presentazione dell'architettura definita e di tutti i suoi componenti, specificandone l'utilizzo effettuato in questo caso specifico.

- *Capitolo 4* - Si occupa di presentare il datset sul quale è stato sviluppato lo Use Case di *Predictive Maintenance*. Verranno forniti dettagli in merito a come si compone il dataset, sia dal punto di vista dei veicoli che delle informazioni acquisite, evidenziando alcune problematiche che lo affliggono.

- *Capitolo 5* – Presenta le famiglie di tecniche di *Machine Learning* valutate per lo sviluppo dello Use Case di *Predictive Maintenance*. Per ognuna di esse verrà fornita una breve descrizione, utile a giustificare se e perché quella tipologia di approccio sia stata applicata o meno.

- *Capitolo 6* – Con questo capitolo si entra nel cuore dello sviluppo dello Use Case di *Predictive Maintenance*. Verranno fornite alcune informazioni in merito alle scelte di sviluppo, per poi passare alle elaborazioni eseguite sul dataset, tralasciando le operazioni di *pre-processing* standard, a favore di quelle *custom*. Infine, ci si addentrerà, in via definitiva, nello sviluppo degli algoritmi di apprendimento automatico, supervisionato e non, passando per lo sviluppo di metodi utili e per la definizione della logica di etichettatura dei dati.

- *Capitolo 7* – Chiude la presentazione dello sviluppo dello Use Case, fornendo tutti i risultati del caso, partendo da quelli dei metodi utili, per arrivare al risultato prodotto dall'algoritmo di *clustering*, ovvero K-means. Successivamente verranno forniti i valori delle metriche di valutazione del modello di classificazione definito tramite XGBoost e, in aggiunta, anche l'accuratezza delle previsioni finali, valutata tramite l'utilizzo di un dataset contenente lo storico di tutti gli interventi di manutenzione, fornito a valle dell'intero processo di sviluppo.

- *Capitolo 8* – A conclusione della presentazione di tutto il lavoro svolto, verranno formulate le conclusioni del caso e saranno forniti alcuni spunti per le possibili evoluzioni di questo Use Case e lo sviluppo di altre attività di valore per il cliente.

CAPITOLO 1

Panoramica dell'impresa automobilistica: unità, esigenze e obiettivi di business

In questo capitolo iniziale verrà presentata l'azienda cliente di Deloitte Consulting. Verrà proposta una overview sulla struttura interna, con una breve presentazione delle diverse divisioni che la compongono, per poi focalizzarsi sulla strategia che guida le attività aziendali, sugli obiettivi e sulle necessità di business evidenziate a livello generale.

1.1 Premessa

L'azienda cliente di Deloitte Consulting della quale tratterà questo lavoro di tesi è una casa automobilistica italiana che vanta decenni di attività durante i quali ha sempre prodotto delle vere e proprie auto da sogno. In oltre mezzo secolo, l'azienda è stata anche in grado di allacciare saldi rapporti con il mercato estero, rendendo il marchio noto e desiderato in tutto il mondo. Fortunatamente ancora oggi il cuore pulsante di questa casa automobilistica, partendo dalla produzione e arrivando al suo business, batte ancora in Italia, rendendo noi tutti orgogliosi di un *Made in Italy* di così tanto pregio e rispetto.

In grado di ascoltare e comprendere le esigenze del cliente, Deloitte Italia è arrivata a garantirsi un posto di partner irrinunciabile, grazie soprattutto ai team che, negli anni, si sono succeduti nell'affiancare il lavoro di questa azienda. La profonda conoscenza del settore maturata negli anni, ad oggi consente alla stessa Deloitte di farsi promotrice di nuove proposte di sviluppo che si sono sempre rivelate efficaci e di effettivo supporto alle scelte di business, tanto da portare ad osservare anche dei sensibili aumenti nelle vendite.

1.2 Struttura organizzativa – Unità di Business

La struttura organizzativa della società è di tipo funzionale con meccanismi di integrazione attraverso comitati "*ad hoc*". Negli organigrammi aziendali sono rappresentate le diverse funzioni che riportano direttamente al CEO della società, insieme alle altre unità organizzative di cui si compone l'attività aziendale, le linee di dipendenza gerarchica e funzionale tra le stesse e i nomi dei soggetti che occupano le varie posizioni organizzative.

Nelle prossime sottosezioni verrà fornita una presentazione delle diverse unità aziendali.

1.2.1 Research &Development

Ricerca, innovazione, evoluzione e sperimentazione sono linee guida dell'intera azienda, tutte racchiuse in questa imprescindibile unità funzionale il cui compito è quello di tradurre le idee in prodotto. È qui che il segno lasciato dalla matita prende forma e diventa metallo, fibra di carbonio, pelle, elettronica, potenza e piacere di guida.

Le macroaree di cui si compone sono: Powertrain, BIW – Trim, Whole Vehicle Development, Chassis, Advanced Composites and Lightweight Structure Development, R&D Project Management, Design, Electric, Electronic Car Development, Concept Development e Motorsport.

1.2.2 Finance

Si occupa di supportare le diverse aree aziendali in tutte quelle decisioni che hanno rilevanza economica o che riguardano la pianificazione e il conseguimento dei risultati, presenti e futuri, nel rispetto delle procedure e delle regole stabilite.

È un vero e proprio network costituito da diverse macroaree: Administration, Controlling, IT, Tax, Legal, Compliance e Risk Management.

1.2.3 Product Marketing & Communication

Si occupa di definire la strategia di sviluppo del marchio con l'intento di renderlo il primo driver della crescita dell'azienda, di gestire tutte le attività di Licensing e Category Management, di Sponsorship e le visite degli ospiti VIP.

Nell'ottica di sviluppare nuovi e attraenti servizi è anche fondamentale conoscere esigenze e desideri del cliente; l'innovazione, come dicevamo, è infatti uno dei valori fondanti dell'azienda e ispira le attività di *branding* e comunicazione rivolte alla *community* di possessori delle sue vetture.

All'interno di Product Marketing & Communication possiamo riconoscere un'ulteriore suddivisione in Marketing Operations, Brand Extension e Comunicazione.

1.2.4 Commercial

Il dipartimento Commercial ha la *mission* di garantire l'eccellenza qualitativa di prodotti e servizi forniti. In particolare, si occupa di servizi di vendita, ma anche e soprattutto post-vendita, fondamentali per assicurarsi di far vivere al cliente una vera e propria *Luxury Experience*. È, quindi, una unità che, al suo interno, coinvolge tutta la rete di vendita, le cui mansioni spaziano fino ad arrivare ad affiancare Product Marketing & Communication nella cura dell'immagine del *brand* attraverso i canali di contatto con clienti e fan.

Le macroaree che compongono questa unità sono: Marketing Prodotto, Retail Marketing & Customer Journey, CRM, Sales Operations and Planning, After Sales e Franchise and Business Development.

1.2.5 Strategy & Connectivity

È un dipartimento che al proprio interno racchiude tre differenti unità: Product Strategy, Process & Methods e Connectivity.

La parte strategica si occupa di prioritizzare le attività, di gestire e monitorare il budget, ma anche di far approvare le richieste al consiglio di amministrazione. A livello di processi si studiano metodi e strumenti atti a migliorare i prodotti esistenti e si definiscono le *best-practice* per le varie linee di produzione. Il ramo di Connectivity, invece, definisce e implementa i servizi connessi per tutto il portfolio aziendale. In aggiunta a questo, per garantire sempre la

massima avanguardia su tutti i fronti, si occupa di effettuare un continuo *scouting* alla ricerca di novità tecnologiche da poter sfruttare nella progettazione dei nuovi veicoli.

1.2.6 Human Capital & Organization

Human Capital & Organization ha il compito di contribuire allo sviluppo strategico del business, dell'azienda e delle sue persone. Le persone, infatti, sono al centro dell'organizzazione, poiché sono i valori che esse esprimono nel loro lavoro a garantire il successo dell'intera azienda.

Le macroaree sono: HR Operations, Industrial Relations, Corporate Security & General Services e Organization.

1.2.7 Production

È dove le capacità manifatturiere dell'azienda si mostrano al loro meglio, arrivando alla concreta realizzazione delle vetture che tutti gli amanti del lusso, dello stile e delle supercar sognano. Il compito del team di produzione è quello di industrializzare un'emozione e di costruirla, preservandone l'essenza di genialità e unicità.

Le macroaree sono: Safety & Environment, Technology, Logistica, Produzione, Centro materiali compositi (CFK) e Production Engineering.

1.2.8 Procurement

Questo dipartimento si occupa della gestione degli approvvigionamenti aziendali, partendo dalle materie prime fino ai macchinari e alle attrezzature necessarie per la produzione. Tra i loro compiti chiaramente ricade anche quello di individuare i fornitori, valutare i rischi associati, contrattare sui prezzi e vigilare sulla puntualità delle consegne.

Internamente si divide nelle seguenti macroaree: Gestione Approvvigionamenti, Risk Management e Cost Analysis.

1.2.9 Quality

Ha la *mission* di garantire che i prodotti superino sempre le aspettative del cliente in termini di affidabilità e qualità. Il lavoro di questa divisione si fonda su due principi fondamentali: miglioramento continuo e orientamento agli standard di qualità aziendali. Gli obiettivi di qualità sono definiti per aree, processi e progetti e prevedono la produzione di automobili affidabili e dal fascino inimitabile, un'assistenza eccellente e processi razionali, ecologici e sicuri.

Quality è suddivisa nelle seguenti macroaree: Qualità di Produzione, Qualità Fornitori, Qualità Preventiva e Qualità Analisi.

1.3 Strategia aziendale

L'intero mondo dell'automobile sta mutando sempre più velocemente; negli anni, infatti, abbiamo assistito a un progressivo aumento delle tecnologie di bordo e dei dispositivi di assistenza alla guida che hanno reso i veicoli dei veri e propri dispositivi IoT. Non da ultima, la sempre più crescente attenzione verso l'ambiente rilevata a livello globale ha portato perfino alla sostituzione dei motori termici a favore di quelli elettrici, ritenuti più *green*. Una frontiera che, probabilmente, solo 15 anni avremmo ritenuto invalicabile, ovvero pura immaginazione.

Proprio per far fronte a questi cambiamenti e non ritrovarsi impreparata, già nel 2017 l'azienda ha definito un'ambiziosa strategia aziendale che si protrarrà fino al 2025, in grado di evidenziare obiettivi e priorità a lungo termine, definire chi vuole essere nei prossimi anni e decidere come interpretare i nuovi trend che caratterizzeranno sempre di più il mondo dell'automobile in futuro, in particolare dal punto di vista della sostenibilità, digitalizzazione e urbanizzazione.

Nello stesso documento che dichiara la strategia, viene anche definita la *vision* aziendale, che risulta essere molto chiara ad ogni dipendente e collaboratore: entro il 2025 l'azienda vuole imporsi sul mercato diventando l'icona delle automobili supersportive di lusso, e la stretta collaborazione con Deloitte, iniziata proprio poco dopo il 2017, ne è prova e strumento.

Le parole chiave che determinano il modo con il quale si vogliono raggiungere tutti gli obiettivi prefissati sono sviluppo, produzione, avanguardia ed esperienza del marchio, affinché questo possa diventare fonte di entusiasmo e ispirazione non solo per i clienti ma anche per i semplici appassionati. Essere leader di mercato, infatti, significa anche essere in grado di realizzare prodotti all'avanguardia, che offrano il meglio del design, emozione, performance, innovazione e qualità; tali capacità si sposano alla perfezione con questa casa automobilistica.

1.4 Obiettivi aziendali

A supporto della strategia aziendale, sono stati definiti anche degli obiettivi misurabili, come l'aumento del margine di profitto e dei volumi di vendita, il miglioramento dell'attrattività del marchio e della sostenibilità. Se i primi tre sono obiettivi più ovvi e comuni ad ogni azienda, anche al di fuori di questo specifico mercato, la quarta è sicuramente molto più in linea con il periodo storico che stiamo vivendo e con le sfide che siamo chiamati ad affrontare a livello globale, ma è, allo stesso tempo, anche più inaspettata, considerando, comunque, che stiamo parlando di un'azienda il cui prodotto finale è un veicolo a combustione termica.

Proprio a questo proposito, l'azienda si è posta l'obiettivo di perseguire un business che sia sostenibile anche dal punto di vista ambientale, non solo in termini di riduzione delle emissioni della propria flotta, ma anche di contenimento e compensazione delle emissioni di CO_2. Al fine di continuare ad aggiudicarsi la certificazione CO_2 neutrale, una sfida enorme e rispetto alla quale l'azienda sente una profonda responsabilità, viene investito molto impegno nella continua ricerca e sviluppo di tecnologie all'avanguardia e nei processi virtuosi, che permettano di limitare l'impatto, evitare sprechi, contenere i consumi e prevenire l'inquinamento ambientale.

1.5 Necessità di Business

Oggi tutto ciò che facciamo, in particolar modo a livello informatico, si traduce nella produzione di una serie di dati che navigano la rete e vengono immagazzinati nei server. Basta considerare la quantità di operazioni svolte in ogni istante in tutto il mondo per capire la portata del fenomeno, che prende il nome di Big Data, e giustificare perché esso abbia introdotto l'immediata necessità di nuove tecnologie e processi in grado di immagazzinarli, gestirli e sfruttarli opportunamente.

Le aziende automobilistiche non sono di certo immuni a questa ondata di cambiamento. Anzi, per certi versi, devono anche gestire quantità di dati ancora più elevate, in quanto aziende produttive si trovano a dover immagazzinare i dati provenienti dai macchinari, ma anche quelli prodotti a valle delle attività quotidiane dei vari dipartimenti, ai quali si aggiungono quelli inviati dai veicoli venduti che, grazie alle tecnologie e ai sensori di bordo, fungono da veri e propri dispositivi IoT.

Fondamentalmente, quindi, uno dei bisogni principali dell'azienda è quello di gestire i Big Data a sua disposizione, comunemente descritti dalle cosiddette "*5 V*", elencate qui di seguito:

- *Volume*: grandi volumi di dati con crescita esponenziale, caratterizzati anche da un alto livello di dettaglio.
- *Varietà*: i dati provengono da diverse fonti e sono eterogenei, quindi di vario tipo: da quelli strutturati e numerici nei database relazionali, a quelli non strutturati (come i documenti di testo) e semi-strutturati (come e-mail, video, audio, dati di stock e transazioni finanziarie, etc).
- *Velocità*: i dati vengono ricevuti e devono essere gestiti in modo tempestivo e a una velocità senza precedenti.
- *Veridicità*: i dati provengono da tante fonti diverse, per cui risulta difficile, ma necessario, pulire, trasformare e identificare le relazioni tra le informazioni provenienti dalle diverse sorgenti per poterle integrare e utilizzare.
- *Variabilità*: il significato o l'interpretazione di uno stesso dato può variare in funzione del contesto in cui questo viene raccolto ed analizzato. Il valore, quindi, non risiede soltanto nel dato, ma è strettamente collegato al contesto da cui si ricava.

Limitarsi a raccogliere i dati, pur sfruttando le migliori tecnologie disponibili sul mercato, non garantisce, però, di avere informazioni e, soprattutto, di trarne vantaggio estraendo conoscenza. Proprio per questo motivo potremmo aggiungere un'ulteriore *V* che giustifichi il crescente interesse delle aziende nei confronti dei Big Data: *Valore*. Estrarre valore dai dati, quindi conoscenza, è ciò che fa davvero la differenza, è il motore di quello che ormai è diventato un vero e proprio business e che, da qualche anno a questa parte, costituisce un vantaggio competitivo che consente alle aziende, che ne fanno uso in maniera efficace, di porsi sul mercato con una marcia in più.

Per generare valore a partire dai dati si utilizzano opportuni strumenti e tecniche di analisi che consentono di estrarre informazioni di valore da poter, poi, reinvestire per ottimizzare i processi interni, oppure a supporto dei processi di *decision making* aziendale, in modo da assumere decisioni più rapide e, soprattutto, informate. Nello specifico, la casa automobilistica è interessata ad aprirsi anche al mondo della *Data Science* allo scopo di poter eseguire analisi più accurate tramite tecniche e algoritmi di *Advanced Analytics*. Queste, infatti, possono aiutarla ad identificare *trend* e *pattern* per migliorare e personalizzare le attività di marketing, a reperire informazioni utili per migliorare la *customer experience* ed eseguire manutenzione predittiva, riducendo, quindi, i tempi di fermo e i costi di manutenzione dei veicoli e dei macchinari e il numero di *failure* nei quali possono incorrere.

L'adozione di metodologie di Big Data espone però l'azienda a nuove potenziali minacce; infatti sull'acquisizione e gestione dei dati vigono norme molto restrittive che, qualora non venissero rispettate, porterebbero l'azienda al rischio di incorrere in sanzioni giudiziarie o amministrative piuttosto severe. Tra questi dati, inoltre, sono inevitabilmente presenti anche dati personali e sensibili; basti pensare all'anagrafica dei clienti o dei dipendenti, anch'essi sottoposti a specifiche normative atte a regolare la tutela della privacy degli individui. In ambito europeo, è il GDPR, che dal 2018, norma tutti i comportamenti da seguire per condurre un'acquisizione e gestione dei dati corretta e legale all'interno del territorio dell'Unione Europea.

CAPITOLO 2

Piattaforma dati aziendale e Data Science

Con questo capitolo si entra nel vivo delle necessità di business. Verranno innanzitutto fornite informazioni in merito alle attuali attività condotte da Deloitte per questo cliente, con riferimento all'architettura dati implementata all'inizio della collaborazione e tuttora utilizzata. In secondo luogo, evidenziando l'inadeguatezza di questa struttura per le nuove esigenze di business, verrà introdotta una tipologia di ambiente di archiviazione dei dati più idonea a rispondere ai nuovi bisogni interni, ovvero il Data Lake. Lo sviluppo di questa nuova architettura consentirà di mettere in campo analisi più spinte in grado di estrarre ancora più valore dai dati a disposizione, così come richiesto lato Business. Nella parte conclusiva del capitolo saranno, quindi, presentati una serie di Use Case, per lo più di Big Data e Advanced Analytics, che sono stati delineati e che, una volta resi disponibili, forniranno un supporto all'azienda sia lato strategico, sia nel miglioramento delle attività quotidiane. Tanto per l'implementazione architetturale quanto per lo sviluppo degli Use Case verranno indicati gli approcci definiti e seguiti da Deloitte.

2.1 Premessa

Dal 2018 ad ora Deloitte si è sempre occupata di condurre attività di *Business Intelligence* per conto di questo cliente. Negli anni sono stati prodotti centinaia di *Documenti*, *Dossier* e *Report* su *MicroStrategy* in grado di rispondere ai bisogni interni dell'azienda, evidenziando informazioni estraibili dai dati mediante analisi principalmente di tipo descrittivo, sulle quali poter ragionare per prendere decisioni più ponderate e consapevoli a livello strategico-manageriale.

Per le attuali attività di *Business Intelligence*, tutti i dati sono contenuti all'interno di un Data Warehouse aziendale aggiornato con frequenza giornaliera. Questo tipo di base di dati non è, però, adatto alla conservazione dei Big Data che, per loro natura, provengono da sorgenti varie ed eterogenee, e per i quali non si è in grado di predeterminare uno schema logico. Nasce quindi, la necessità di costruire un nuovo ambiente di archiviazione centralizzata dei dati, ovvero il Data Lake.

Questa evoluzione architetturale, all'apparenza strettamente tecnologica, può, in realtà, essere il principale elemento abilitante per lo sviluppo di una cultura aziendale *data-driven* e per la realizzazione di analisi più sofisticate, le cosiddette *Advanced Analytics*, atte a scoprire relazioni più profonde nascoste nei dati, alle quali la casa automobilistica è interessata per via del potenziale applicativo del valore ricavabile.

2.2 Piattaforma dati aziendale: Data Lake

Il termine Data Lake si deve a James Dixon, CTO Pentaho, che, nello spiegare il concetto in confronto ai Data Mart che compongono un Data Warehouse, disse:

> *"Se si pensa a un Data Mart come riserva di acqua in bottiglia, pronta ad un immediato consumo, il Data Lake è una grande massa d'acqua in uno stato più naturale. I contenuti del lago fluiscono dalle sorgenti e vanno a riempire il lago; gli utenti possono venire a esaminare le acque, tuffarsi o prelevare campioni"*

Sostanzialmente, quindi, un Data Lake è un archivio centralizzato nel quale i dati, come l'acqua, fluiscono liberamente, in real-time o in batch, dalle numerose fonti che li generano. I dati prodotti dai sistemi aziendali e dai veicoli connessi vengono, quindi, conservati nel loro formato originale, siano essi strutturati, semi-strutturati o non strutturati, senza necessità di riconciliarne le eterogeneità. Si potrebbe, quindi, parlare di modello di dati con *schema-on-read*. D'altra parte, il Data-Lake permette di strutturare i dati quando li si recupera dallo *storage*; in questo modo, quindi, è solo nel momento in cui si ha la necessità di utilizzarli per uno specifico Use Case che i dati subiscono un processo di trasformazione che li porterà ad assumere lo stato più idoneo alle successive fasi di analisi

Come per ogni cosa, i vantaggi introducono anche delle complicazioni; in questo caso, la sfida principale con un archivio di questo tipo consiste proprio nel fatto che i dati non elaborati vengano archiviati senza alcuna supervisione dei contenuti; quindi, affinché questi possano poi essere effettivamente utilizzabili in futuro, è necessario avere degli specifici meccanismi di *governance*, atti a catalogarli e proteggerli. In caso contrario, infatti, non sarà possibile reperire i dati o reputarli affidabili, rendendo, quindi, il Data Lake una struttura inutilizzabile che prende il nome di *Data Swamp*, ovvero una palude di dati, non in grado di rispondere alle reali esigenze del cliente.

Allo stato dell'arte i primi Data Lake erano costruiti su *cluster HDFS on-premise*; ad oggi, però, esistono numerose soluzioni di *Infrastructure-as-a-Service* che portano a prediligere la soluzione cloud. La soluzione locale ha, infatti, una serie di svantaggi:

- esigenza di gestire sia l'infrastruttura hardware che il lato software, con conseguente necessità di figure professionali con conoscenze in entrambi gli ambiti;
- all'investimento iniziale necessario per l'acquisto degli apparati HW, si aggiungono e si protraggono nel tempo i costi di gestione, funzionamento e manutenzione, comprensivi dei costi degli impianti di condizionamento e di quelli immobiliari;
- necessità di gestire manualmente la scalabilità del Data Lake per consentire l'uso ad un maggior numero di utenti o aumentare le capacità di *storage*.

Al contrario, il *cloud computing* consiste nella distribuzione *on-demand* delle risorse IT tramite Internet, con una tariffazione basata sul consumo. Piuttosto che acquistare, possedere e mantenere i *data center* e i server fisici, è possibile accedere a servizi tecnologici, quali capacità di calcolo, *storage* e database, sulla base delle proprie necessità. Complessivamente, quindi, la soluzione cloud offre una serie di importanti vantaggi:

- eliminaazione della necessità di costruire e mantenere l'infrastruttura HW, riducendo, quindi, i costi di acquisti, progettazione e manutenzione ai soli costi di effettivo utilizzo del servizio;
- gestione della scalabilità a carico del provider del servizio di IaaS;
- elasticità, che permette di gestire la quantità di risorse a seconda delle necessità;

- flessibilità dei servizi cloud che offrono un'infrastruttura agile, facilmente modificabile e adattabile al soddisfacimento di altri Use Case;
- disponibilità di tecnologie sempre aggiornate;
- altissima affidabilità e disponibilità dei servizi;
- servizi disponibili in varie regioni, aspetto di fondamentale importanza, considerando che l'azienda cliente ha delle sedi anche all'estero, e avvicinare le applicazioni agli utenti finali riduce la latenza.

Valutati i pro e contro di entrambe le possibilità, la casa automobilistica, consigliata da Deloitte, ha ritenuto più opportuno e vantaggioso optare per una soluzione cloud e, considerati i principali vendor disponibili, la scelta è ricaduta su Amazon Web Services (AWS). Nel Capitolo 3 verrà presentata l'architettura progettata e i servizi che la compongono.

2.2.1 Approccio Deloitte

2.2.1.1 Framework Architetturale

Grazie all'esperienza maturata negli anni su più clienti e mercati eterogenei, Deloitte ha delineato un *Framework* architetturale, riportato in Figura 2.1, adattabile anche a questo contesto.

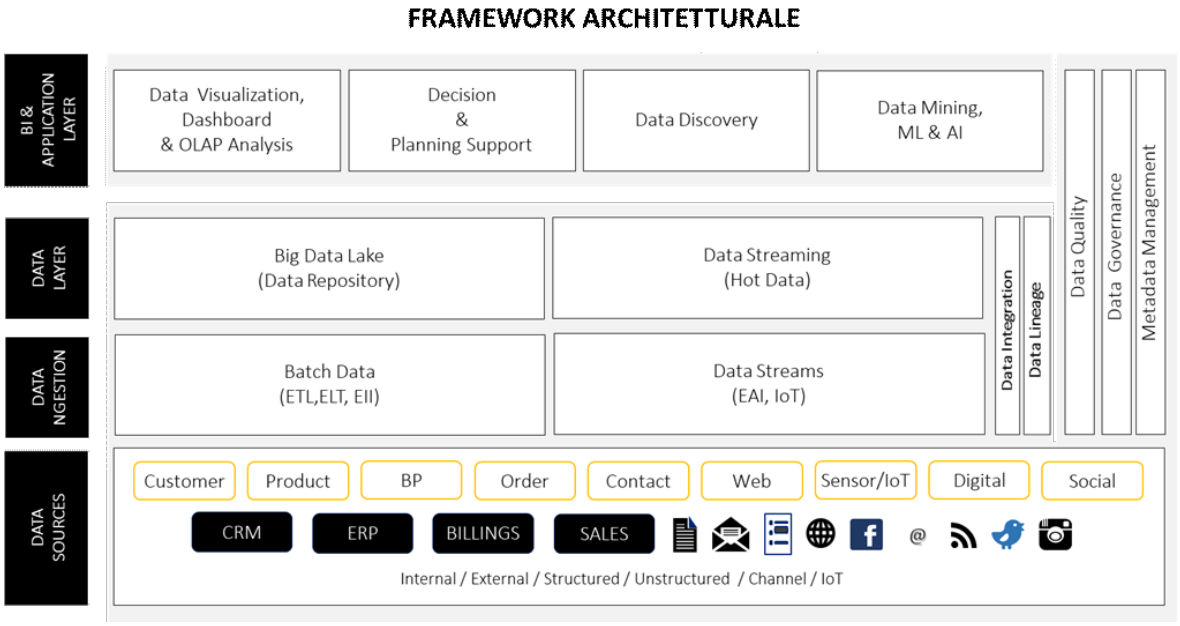

Figura 2.1: Framework architetturale di Deloitte

Qui di seguito verranno descritti gli elementi che compongono il *framework*:

- *Sorgenti Dati:* rappresenta fonti di dati interne, cioè sistemi aziendali, ma anche esterne, come social network, servizi web, IOT e altro. Poiché le fonti stesse sono altamente eterogenee, non ci sono restrizioni sulla natura strutturata dei dati, né sui volumi o la velocità di ricezione.
- *Data Ingestion Layer:* rappresenta il processo di acquisizione e preelaborazione dei dati in modalità *Batch* o *Near Real-Time*. Questi dati possono essere acquisiti attraverso

processi ETL/ELT standard (*Extract-Load-Transform*, *Extract-Transform-Load*), *Enterprise Information Integration* (EII) o flussi *Near Real-Time* ottenuti tramite tecniche NRT (*event driven* o *Micro-Batch*), *message bus* (EAI). In alternativa essi possono provenire dai sensori montati sui dispositivi IoT.

- *Data Layer*
  - *Big Data Lake (Data Repository):* questo layer prevede la ricezione dei dati in una *Landing Zone* e permette la loro elaborazione in formato grezzo (*Raw Data Zone*), rispecchiando il modello dei sistemi di origine. Durante questa fase si applicano, anche, i processi di Data Quality e Validazione (*Data Quality Zone*). L'obiettivo finale è quello di integrare i dati provenienti da fonti interne ed esterne, per offrire viste uniche e integrate di tutte le unità di business (*Refined Data Zone*) e consentire agli utenti di esplorare i dati, cercando correlazioni o altri scenari di Analisi e *Data Discovery*, volti a identificare o convalidare nuove possibili regole di business.
  - *Data Streaming (Hot Data):* gestisce l'elaborazione e l'arricchimento dei dati in tempo reale abilitando capacità di *Streaming Analytics* e *Complex Event Processing*. Le soluzioni di elaborazione dei flussi sono progettate per gestire grandi volumi di dati in tempo reale, con un'architettura scalabile.
  - *Data Integration:* permette di progettare e orchestrare l'elaborazione inter-piattaforma e quella intra-piattaforma. Avendo a disposizione una *Big Data Platform*, è possibile integrare grandi quantità di dati di qualsiasi formato e da qualsiasi fonte. Il componente *Lineage* permette la comprensione del ciclo di vita dei dati, facilitando lo sviluppo di nuovi processi di elaborazione.
- *AI & Application Layer*
  - *Data Visualization, Dashboard & OLAP Analysis:* supporta l'esposizione dei dati attraverso *report*, cruscotti e tecniche di visualizzazione avanzata, per identificare i fenomeni di business. Fornisce anche agli utenti le tradizionali capacità di *Business Intelligence* e *Self-Reporting*.
  - *Data Discovery:* fornisce strumenti per il *Data Profiling*, la ricerca automatica di correlazioni e modelli nei dati (strutturati e non strutturati). Supporta attività elementari di analisi dei dati per *Data Science* e *Data Governance*.
  - *Data Mining, ML & AI:* fornisce strumenti per eseguire il *preprocessing* dei dati, l'estrazione degli stessi e gli algoritmi di *Machine Learning*. Questi strumenti sono, infatti, in grado di sfruttare la potenza di elaborazione dei sistemi di *Data Storage* per processare gli algoritmi.
  - *Decision & Planning Support:* fornisce un supporto decisionale per gestire le molteplici situazioni che si presentano, sia a livello periferico che centrale, nella gestione dei progetti, al fine di identificare i fenomeni critici e pianificare/ripianificare le attività. Le attività principali consistono nell'analisi delle caratteristiche dei fenomeni critici per la loro riduzione, nonché nell'analisi del profilo strategico e operativo, evidenziando punti di forza e di debolezza, e nella successiva mappatura periodica della tipologia di eventi critici individuati.
- *Data Quality:* fornisce capacità di convalida dei dati e di monitoraggio della qualità.
- *Data Governance e Metadata Management:* La *Data Governance* supporta l'implementazione dei modelli di *governance* e il loro monitoraggio e, in collaborazione con il componente *Metadata Management*, gestisce l'integrazione dei metadati di business con quelli tecnici

e operativi. Un'efficace *Data Governance* richiede la definizione di parametri per la gestione e l'uso dei dati, la creazione di processi per risolvere i problemi ad essi legati e l'abilitazione di decisioni basate su dati di alta qualità e risorse informative ben gestite.

2.2.1.2 Architettura Lambda

In questo caso specifico la scelta è quella di implementare il Data Lake con una architettura di tipo Lambda (Figura 2.2).

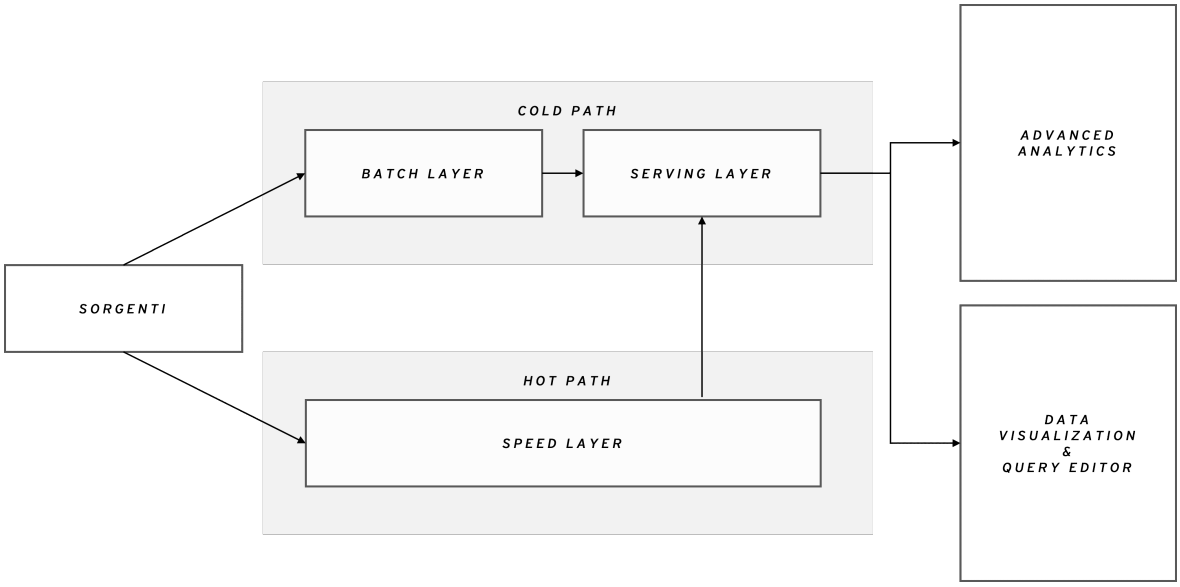

Figura 2.2: Architettura Lambda

L'architettura Lambda è un'architettura progettata per elaborare grandi volumi di dati, gestire letture a bassa latenza e aggiornare grandi quantità di dati in modo scalabile e *fault tolerant*. Permette di acquisire una sequenza immutabile di record e di inserirla in due flussi di elaborazione paralleli, ovvero il sistema *Batch* e quello *Real-Time*, permettendo, così, la realizzazione di casi d'uso in cui è necessario combinare i dati acquisiti in *Real-Time* con quelli elaborati in modalità *Batch*.

Tutti i dati che entrano nel sistema, possono, quindi seguire due diversi percorsi di elaborazione:

- *Cold Path*, chiamato anche *Batch Layer*, in cui l'*ingestion* dei dati viene eseguita in modalità Batch con frequenza predefinita. Memorizza tutti i dati in entrata nel formato sorgente ed esegue l'elaborazione utilizzando un sistema distribuito, in grado di gestire quantità molto grandi di dati. Il risultato di questa elaborazione viene memorizzato in un database di sola lettura, con aggiornamenti che sostituiscono completamente le viste *Batch* preesistenti.

- *Hot Path*, abilitato dallo *Speed Layer*, in cui l'*ingestion* dei dati viene eseguita in modalità *Real-Time*. Questo layer è usato per definire le viste in tempo reale, senza la necessità di garantire coerenza o completezza dei dati. Lo *Speed Layer* mira a completare le viste *Batch* riempiendo il "vuoto" causato dal ritardo del *Batch Layer* nel fornirle.

I dati elaborati nel Cold e nell'Hot Path confluiranno nel *Service Layer* che risponderà a *query* ad hoc restituendo viste pre-elaborate e precompilate. Tutti i dati nel *Service Layer* sono accessibili dal *Data Visualization & Query Layer* attraverso strumenti di visualizzazione dei

dati per la creazione di *report* e *dashboard*, e dall'*Advanced Analytics Layer*, per lo sviluppo di modelli di *Machine Learning* e *Predictive Analytics*.

I principali vantaggi dell'utilizzo di questo tipo di architettura possono essere riassunti come segue:

- gestione di grandi quantità di dati in modo efficiente;
- possibilità di applicare algoritmi avanzati sia *Real-Time* che in *Batch Mode*;
- disponibilità di una vista unificata che combina *Batch* e *Speed Layer*;
- possibilità di applicare algoritmi di *Machine Learning* in modo esteso nel *Batch Layer*, e in modo coerente nello *Speed Layer* e nel *Serving Layer*;
- mantenimento dei dati di input invariati;
- possibilità di esecuzione simultanea *Batch/Real-Time* e aggregazione automatica dei dati.

2.3 Data Science

L'approccio all'*Enterprise Data Platform* attraverso l'implementazione di Data Lake abilita la possibilità di aprirsi al mondo della Data Science attraverso analisi statistiche, descrittive, diagnostiche, predittive e prescrittive. I Big Data prodotti dalle sorgenti a disposizione e, ancor di più, le analisi che possono essere condotte su di essi, consentiranno di sfruttarli e convertirli in intuizioni e conoscenza, in grado di fornire concreto supporto a varie attività aziendali, quindi non solo quelle strategiche.

2.3.1 Approccio Deloitte

Anche in questo caso Deloitte ha delineato un *Framework* per la Data Science, i cui componenti permettono di implementare gli algoritmi di *Machine Learning* da applicare ai Big Data a disposizione. Il *Framework* consiste in 6 step (Figura 2.3) che saranno presentati più in dettaglio nelle prossime sottosezioni.

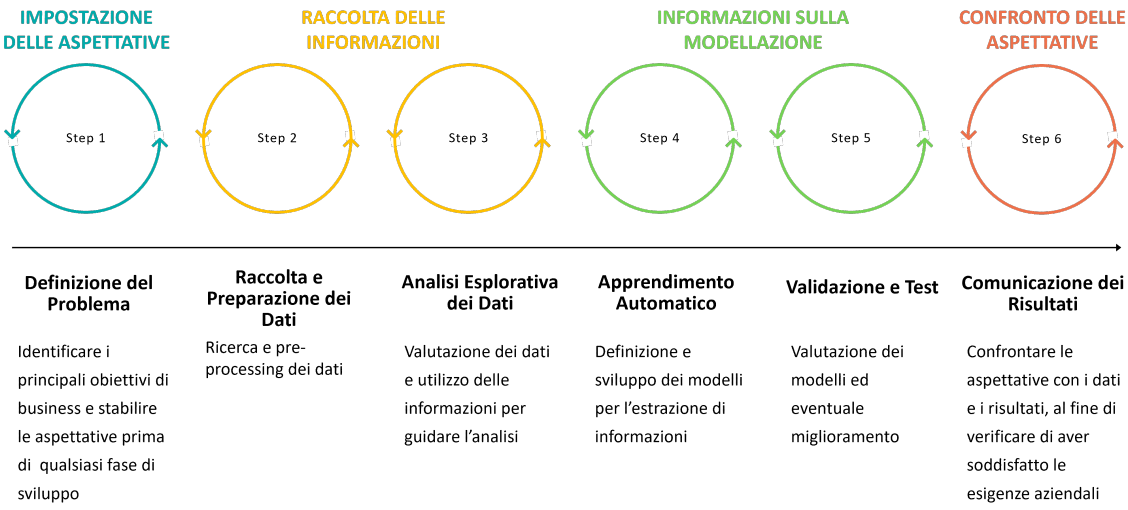

Figura 2.3: Fremework di Data Science seguito da Deloitte

2.3.1.1 Definizione del problema

Per raggiungere gli obiettivi aziendali desiderati, la cosa più importante è definire le esigenze e lo *scope* del progetto. Affinché ciò sia possibile, gli *Stakeholder* dovrebbero condividere la loro conoscenza dei processi coinvolti ed essere allineati sulle principali linee guida del progetto, in modo da definire accuratamente le aspettative per gli esiti e i risultati e trasmettere ai Data Scientist la conoscenza dell'area di Business.

In questa fase, tra le altre cose, occorre eseguire una valutazione della fattibilità del progetto, tenendo conto dell'obiettivo finale, dei dati disponibili e degli eventuali vincoli di bilancio.

2.3.1.2 Raccolta e preparazione dei dati

La preelaborazione e la preparazione dei dati è una delle fasi più importanti e critiche in un progetto di *Data Mining*: i dati grezzi raccolti devono essere processati per essere efficacemente interpretati e analizzati; questa fase, di solito, occupa circa l'80% del tempo complessivo speso nel processo di analisi dei dati.

Per prima cosa si raccolgono i dati, strutturati e non strutturati, dalle varie fonti disponibili; se ne valuta la qualità, tenendo conto anche delle loro eterogeneità, per poterli studiare al fine di capirne il significato e poi pulirli adeguatamente, gestendo eventuali valori mancanti o dati rumorosi, strutturando i dati che non lo sono, etc.

2.3.1.3 Analisi esplorativa dei dati

A questo punto si può eseguire la fase preliminare di estrazione della conoscenza dai dati. Dopo la formulazione di ipotesi preliminari, la visualizzazione grafica dei dati viene utilizzata per cercare pattern, tendenze o relazioni tra le variabili che possano convalidare queste ipotesi. L'uso del tipo adeguato di visualizzazione e il corretto livello di granularità dei dati sono fondamentali per il successo di questa fase di esplorazione degli stessi.

2.3.1.4 Apprendimento automatico

Il cuore pulsante dell'intero processo è, sicuramente, la scelta dell'algoritmo di *Machine Learning*. Con il termine *"Machine Learning"* si fa riferimento ad un insieme di algoritmi addestrati a svolgere un compito specifico appreso direttamente dai dati. Questi si dividono in due macro-famiglie: algoritmi supervisionati e non supervisionati. La scelta tra le due famiglie, e in particolare dello specifico algoritmo che si intende utilizzare, dipende chiaramente dai dati a disposizione e da cosa si vuole ottenere.

Una volta scelto l'algoritmo, bisogna anche occuparsi di definire il modello e scegliere accuratamente i parametri ottimali per un corretto apprendimento.

2.3.1.5 Validazione e test

Definito il modello, è di fondamentale importanza validarlo, ovvero applicarlo ad un insieme di dati di test per i quali è noto l'output desiderato, per poterne valutare l'efficacia e la correttezza predittiva. Confrontando i risultati, è possibile valutare le prestazioni e la qualità del modello utilizzando metriche appropriate, tenendo conto della velocità di elaborazione, dell'errore di generalizzazione, della scalabilità, della facilità di interpretazione dell'output, etc.

Un ulteriore step importante per la validazione del modello prevede il coinvolgimento degli *Stakeholder*; quest'ultimi, infatti, in quanto esperti delle specifiche aree di business interessate dallo Use Case, sono sicuramente le figure più idonee a valutare, realisticamente

parlando, se il modello risponde o meno alle effettive esigenze delineate in partenza e se le previsioni sono corrette. A valle di tali valutazioni, se necessario, è possibile aggiustare il modello per migliorarne le capacità predittive.

2.3.1.6 Comunicazione dei risultati

Alla fine del processo, si sviluppano *Report* e *Dashboard* per condividere e mostrare *insight* e risultati analitici.

La fase di comunicazione dei risultati, sebbene sembri di importanza secondaria, in realtà ha un ruolo assolutamente fondamentale; infatti, affinché il cliente sia in grado di comprendere correttamente e in maniera efficace i risultati e il valore dell'analisi condotta, la corretta scelta degli elementi grafici è sostanziale. Solo in questo modo, infatti, gli *Stakeholder* saranno in grado di valutare quanto il risultato corrisponda ai *desiderata* documentati nella fase di definizione del problema.

2.4 Use Cases

La necessità, espressa a livello aziendale, di iniziare ad operare con tecniche di Data Science più spinte, si riflette chiaramente anche nei bisogni evidenziati internamente dalle singole *Business Unit*.

Il lavoro sinergico tra i rappresentanti delle varie divisioni e il team Deloitte è sfociato nella definizione di una serie di Use Case alcuni dei quali abilitano, a loro volta, la possibilità di svilupparne degli altri, in grado di fornire ancora più supporto alle attività aziendali.

Di seguito verranno riportati gli Use Case, suddivisi per *Business Unit* che ne hanno evidenziato la necessità.

2.4.1 Research & Development

2.4.1.1 D.A.M.A. (Digital Asset Management Assistant)

Attualmente ci si basa sulle esperienze pregresse per pianificare, tramite Excel, l'utilizzo degli *asset* per i test di integrazione e validazione dei componenti elettronici dei veicoli. Le attività da svolgere per ogni progetto sono nell'ordine delle centinaia; quindi gestire possibili ritardi delle attività o non disponibilità degli *asset* è un compito estremamente oneroso.

La richiesta è quella di sviluppare un *tool* di supporto alla pianificazione di utilizzo degli *asset* in questi test, in grado di calcolare il miglior planning, tenendo in considerazione le attività da svolgere e gli *asset* disponibili, e gestire possibili ritardi suggerendo attività alternative.

L'obiettivo è quello di determinare in maniera più precisa la disponibilità e la quantità di *asset* necessari per ogni progetto, in modo da poter pianificare più efficacemente le attività e prioritizzarle meglio, nonché stimare con maggiore accuratezza i costi e i benefici ricavabili dall'acquistare o meno un certo *asset*, in modo da ridurre al minimo gli sprechi di denaro.

Il cliente in questo caso desidera che il tutto venga reso disponibile tramite una web app. Quest'ultima, a fronte dell'inserimento delle varie attività da eseguire, dettagliate con la durata stimata, gli *asset* necessari, le *milestone* da perseguire ed eventuali dipendenze da altre attività, dovrà essere in grado di determinare la migliore pianificazione possibile, fornendola sotto forma di diagramma Gantt, e aggiornarla in caso di ritardo di attività che impattano su quelle di interesse (Figura 2.4).

Come sviluppo futuro ci si aspetta, a seguito di un periodo prolungato di utilizzo dello strumento, la possibilità di sfruttare lo storico delle pianificazioni ottimizzate dal *tool*, come

fonte di informazioni che esso stesso può riutilizzare per effettuare un *fine-tuning* delle pianificazioni, raffinando, quindi, le sue capacità organizzative.

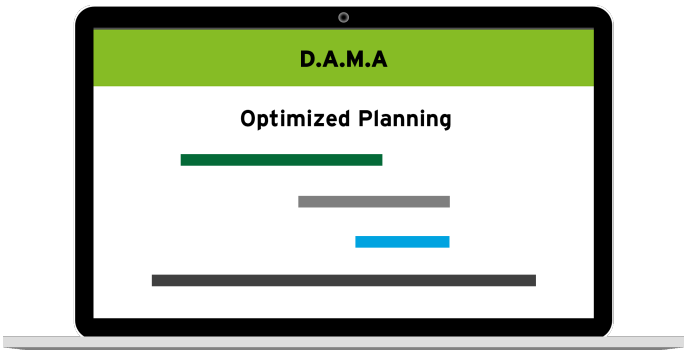

Figura 2.4: Mock-up relativo a D.A.M.A

2.4.1.2 AIV Data Lake

In questo caso il desiderio è quello di definire una porzione del Data Lake aziendale nel quale archiviare e conservare i dati prodotti durante i test eseguiti dal team di Architettura, Validazione e Integrazione elettrica ed elettronica (AIV).

L'output sarà, quindi, un Data Lake esplorabile, contenente i dati delle vetture di prova, inviati in tempo reale, i risultati di ogni test, i *report* stilati dai piloti e i log di quanto accaduto all'interno dell'area di test. Vale la pena sottolineare che i dati che fluiranno nel lago saranno inevitabilmente eterogenei: log, testi, immagini, etc.

Il motivo fondamentale per il quale si vogliono centralizzare queste informazioni è migliorarne la reperibilità, al fine di poterle incrociare e utilizzare nelle analisi condotte sulle risorse materiali, per determinare quali e come vengono utilizzate nelle varie attività, consentendo, quindi, di verificarne la correttezza di impiego. In aggiunta, questa sorgente di dati potrà poi diventare essa stessa fonte di informazioni per ulteriori progetti da sviluppare, sempre in ambito di monitoraggio degli *asset*, tra i quali una versione aggiornata del D.A.M.A. (Sezione 2.4.1.3).

2.4.1.3 D.A.M.A. (Digital Asset Management Assistant) 2.0

In questo caso non si fa altro che migliorare la prima versione del D.A.M.A (Sezione 2.4.1.1) basandosi sui dati raccolti all'interno dell'AIV Data Lake. Pertanto, sia il risultato (Figura 2.4) che gli obiettivi rimangono invariati rispetto alla prima versione del *tool*, con l'aggiunta della possibilità di monitorare l'effettivo impiego degli *asset* durante le attività per le quali ne era stato dichiarato l'utilizzo in fase di pianificazione.

2.4.2 Product Marketing & Communication

2.4.2.1 Funnel Official Site + Salesforce

Per avere una visione d'insieme dell'esperienza utente, che comincia sul web e potrebbe poi sfociare in una reale vendita di una vettura, la richiesta è quella di integrare i dati che tracciano l'esperienza degli utenti che interagiscono con il *form* di contatto sul sito ufficiale con quelli di Salesforce, che traccia tutti gli step successivi all'invio del form, fino all'eventuale acquisto. Salesforce, infatti, è il principale software di CRM al mondo, utilizzato proprio per seguire i progressi dell'utente lungo il *Funnel*, ovvero un modello ad imbuto che, in ambito

marketing, viene utilizzato per descrivere ed analizzare la possibile conversione di un utente in cliente.

Eseguendo algoritmi di *Clustering*, sarebbe possibile identificare dei sottoinsiemi di clienti sulla base dei dati che ne descrivono il comportamento. Tra le categorie di utenti ci si aspetta, in particolar, modo di ritrovare il profilo del fan, che tende a navigare varie pagine del sito ma a non andare oltre la richiesta di informazioni, e quello del *prospect*, che, invece, ha il potenziale di diventare effettivamente un cliente della casa automobilistica.

Nel caso dei clienti inizialmente profilati come *prospect*, che non dovessero però arrivare alla fase di acquisto, sarà interessante anche approfondire le analisi per identificare eventuali colli di bottiglia nel *Funnel*, al fine, se possibile, di risolverli.

Per questo Use Case il cliente si aspetta di poter interagire con una *dashboard* ad hoc, che fornisca una visione d'insieme del percorso degli utenti dal momento in cui entrano sul sito, fino all'ultimo passaggio registrato (Figura 2.5).

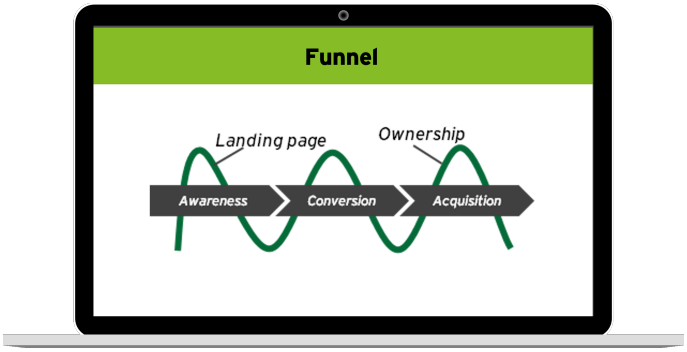

Figura 2.5: Mock-up relativo al Funnel Official Site + Salesforce

2.4.3 Strategy & Connectivity

2.4.3.1 Connectivity Systems Portfolio Monitoring

Ad oggi è compito del dipartimento di Product Marketing analizzare manualmente la penetrazione e l'utilizzo dei servizi di Connectivity, ovvero tutti quei servizi del portfolio dell'azienda legati ai veicoli connessi in rete (veicolo come dispositivo IoT).

Con questo Use Case si vuole spostare l'analisi all'interno del dipartimento dedicato, quindi Strategy & Connectivity, tenendo traccia dei risultati di vendita dei servizi in ciascun paese, considerando vincoli e peculiarità a riguardo. Ove possibile, si dovrà anche monitorare l'utilizzo attivo dei servizi, in modo da poter valutare quali sono quelli più e meno apprezzati.

Queste informazioni dovranno essere rese disponibili attraverso una *dashboard*, che possa essere interrogata per valutare, per ogni servizio, il tasso di penetrazione stimato e una metrica indicante l'utilizzo del servizio in ciascun mercato (Figura 2.6).

Sulla base delle analisi condotte, sarà poi possibile, in primo luogo, prendere decisioni in merito alla dismissione di alcuni servizi, con il vantaggio di un risparmio di risorse da parte dell'azienda, ma anche decidere di sviluppare nuovi servizi o progettare una evoluzione di quelli esistenti che stanno vendendo bene.

2.4.3.2 License Period Optimization

In questo caso il focus è sulle licenze dei servizi attivati dai clienti sui propri veicoli; in particolare, c'è interesse nel determinare il miglior periodo di durata della licenza, tenendo conto dei tempi medi di proprietà di ogni modello e dell'effettivo utilizzo dei servizi. Per poter fare queste valutazioni è, quindi, necessario avere informazioni anche in merito a passaggi

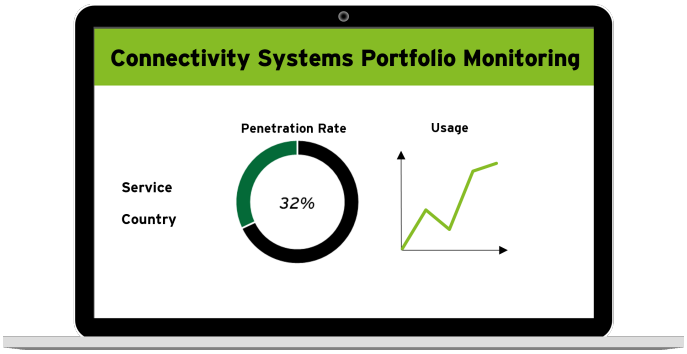

Figura 2.6: Mock-up relativo al Connectivity Systems Portfolio Monitoring

di proprietà dei veicoli, in modo da poter individuare se ci sono dei modelli con tempi di proprietà anomali, quindi particolarmente lunghi o particolarmente brevi, che andranno a costituire degli *outlier* sui quali incentrare ulteriori analisi più approfondite.

L'obiettivo, in questo caso, è quello di assicurare al cliente una durata del servizio che sia più congrua possibile alle esigenze medie, ma, allo stesso tempo, valutare quali sono i servizi utilizzati anche dai proprietari successivi al primo, determinando, quindi, quali siano effettivamente i prodotti a cui i clienti sono più interessati e valutando un'eventuale dismissione degli altri.

Per questa attività l'azienda richiede lo sviluppo di una *dashboard* che, per ogni modello, riporti il tempo medio di proprietà e, per ogni servizio, indichi, invece, il tempo medio di utilizzo attivo (Figura 2.7).

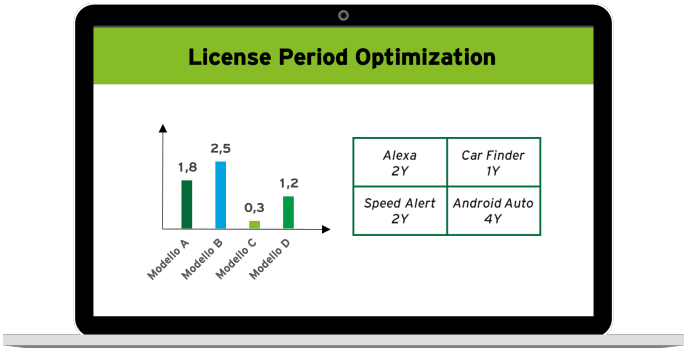

Figura 2.7: Mock-up relativo al License Period Optimization

2.4.3.3 Human Machine Interface User Experience (HMI UX)

Per poter progettare dei miglioramenti per l'interfaccia utente disponibile sul tablet installato a bordo del veicolo, ci si vuole basare direttamente su quella che è l'esperienza dell'utilizzatore del veicolo. L'idea, quindi, è quella di analizzare i percorsi fatti dagli utenti all'interno del menù, per riconoscere quelli standard e le eventuali variazioni (Figura 2.8). In questo modo sarà possibile rilevare potenziali inefficienze delle Human Machine Interface e studiarne un'evoluzione che la renda più fluida e *user friendly*, migliorando così l'esperienza utente.

Lo sviluppo di questo Use Case ha, anche, l'intento di abilitarne un altro che, analizzando i click fatti dall'utente sul tablet, confrontandoli con i comportamenti standard e incrociandoli con le analisi condotte sulle immagini registrate dalle telecamere di bordo, stimi lo stato d'animo del conducente (Sezione 2.4.3.4).

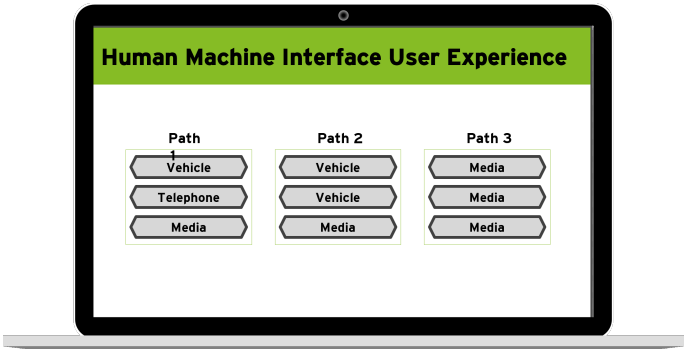

Figura 2.8: Mock-up relativo a Human Machine Interface User Experience

2.4.3.4 Driver Sentiment Analysis

Come già anticipato, l'interesse dell'azienda è anche quello di spingersi oltre, mettendo in campo un tipo di analisi estremamente all'avanguardia, ovvero la *Sentiment Analysis*. Quest'ultima si focalizza principalmente nel riconoscere correttamente gli stati d'animo dell'utente che interagisce con il veicolo, focalizzando la concentrazione su quelli negativi e basando l'analisi sui dati provenienti dalla vettura.

L'idea è quella di sfruttare i dati prodotti dall'interazione dell'utente con il display HMI, incrociarli con quelli derivati dall'analisi sulle immagini inviate dalle telecamere di bordo e confrontarli con quelli registrati in sessioni di guida precedenti. Una volta determinato il profilo di comportamento standard dell'utente rispetto ai dati catturati, sarà possibile, tramite *Anomaly Detection*, identificare delle difformità, come, ad esempio, il click compulsivo sulla stessa icona; queste ultime potrebbero, infatti, essere sintomo di nervosismo dell'utente (Figura 2.9).

L'eventuale rilevazione di un comportamento anomalo potrà, anche, fungere da *trigger* per un sistema di diagnostica che si occuperà di effettuare un controllo generale sullo stato di funzionamento dei principali apparati del veicolo, in particolar modo dell'HMI, in modo da poter individuare un eventuale *fault* che potrebbe essere causa del nervosismo dell'utente, ancora prima dell'apertura del *claim*. In questo modo l'azienda sarà in grado di porsi in maniera proattiva verso le problematiche del cliente, riducendo il lavoro del servizio clienti e migliorando la *user experience*.

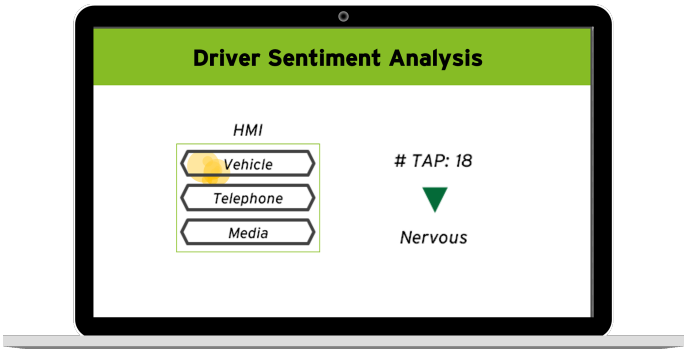

Figura 2.9: Mock-up relativo alla Driver Sentiment Analysis

2.4.4 Production

2.4.4.1 Equipment Predictive Maintenance - PH1

La richiesta, in questo caso, si sviluppa parallelamente su due fronti. In primo luogo, si intende integrare tutti i dati provenienti dalle Isole di Lavorazioni Meccaniche (ILM) all'interno del Data Lake; ad oggi, infatti, solo una parte dei dati provenienti dai macchinari di produzione viene conservata all'interno dei server aziendali. Questa modifica consentirebbe non solo di eseguire un monitoraggio *Real-Time* dei macchinari, ma anche di avere a disposizione una grande mole di dati da utilizzare per *Advanced Analytics* di vario tipo e per il calcolo automatico di alcuni KPI, il principale dei quali è l'*Overall Equipment Effectiveness* (OEE), indice di efficienza delle macchine, attualmente stimato manualmente e, quindi, soggetto anche ad errori umani.

In seconda battuta si vogliono anche analizzare i *warning* provenienti dalle ILM, al fine di identificare pattern che si ripetono nel tempo e che possono essere sintomo di problematiche che si stanno verificando (Figura 2.10). Al momento, infatti, i dati delle ILM vengono integrati nel diario macchina cartaceo, compilato dagli operatori di macchina e usato per effettuare manualmente i primi rilevamenti delle problematiche che potrebbero affliggere l'impianto. L'automatizzazione del processo permetterebbe di individuare quali combinazioni di *warning* possono portare ad errori bloccanti e quali sono i componenti soggetti a maggiore usura o guasto. Alla luce di queste evidenze, sarebbe, quindi, possibile intervenire tempestivamente nella manutenzione delle apparecchiature evitando il fermo macchina, problematica molto diffusa e costosa per l'azienda, garantendo una maggiore efficienza dell'intero impianto produttivo.

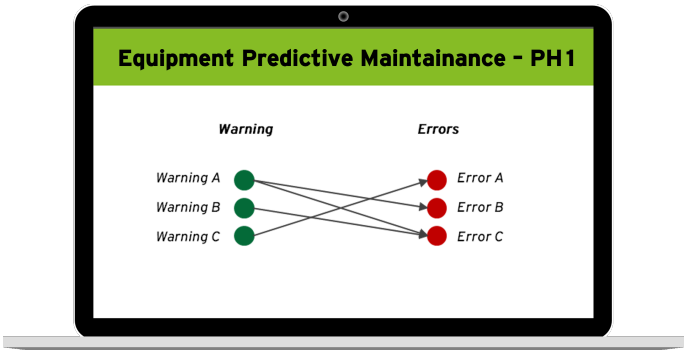

Figura 2.10: Mock-up relativo all'Equipment Predictive Maintenance - PH1

2.4.4.2 Equipment Predictive Maintenance – PH2

Il desiderata è un *report* che mostri per ogni macchina tutti i *warning* associati e, avvalendosi della soluzione progettata nella fase 1 (Sottosezione 2.4.4.1), evidenzi se alcuni di questi corrispondono a modelli ritenuti rischiosi, per i quali è suggerito intervenire con azioni di manutenzione.

2.4.4.3 Equipment Predictive Maintenance – PH3

Il cliente, in questo caso, richiede una web-app che lo supporti nell'individuare le tempistiche più consone all'applicazione di una particolare strategia di produzione per le ILM, evidenziando, anche, i rischi di fallimento associati. Per poter fornire tali informazioni, al di sotto del *report* dovrà operare un algoritmo che simuli gli effetti delle diverse strategie di

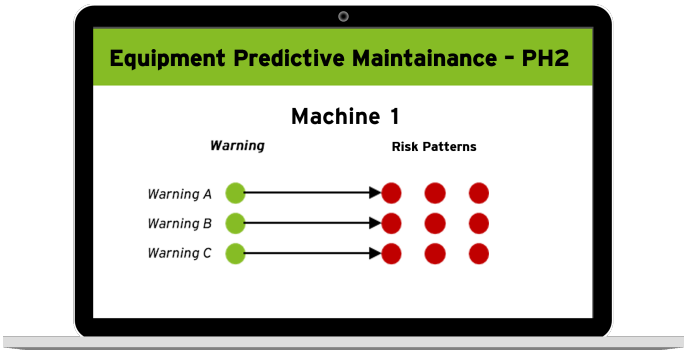

Figura 2.11: Mock-up relativo all'Equipment Predictive Maintenance - PH2

produzione sulle prestazioni delle apparecchiature, in modo da poter determinare rischi e tempistiche con maggiore precisione (Figura 2.12).

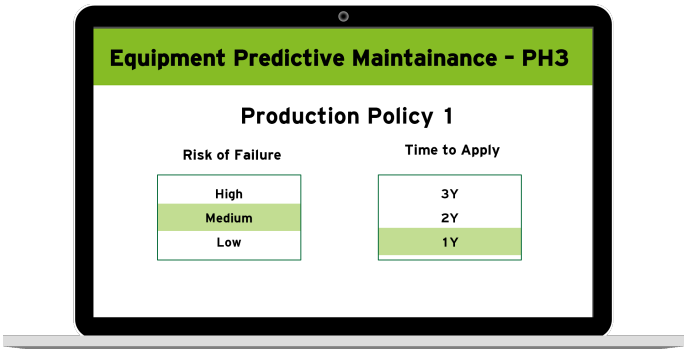

Figura 2.12: Mock-up relativo all'Equipment Predictive Maintenance - PH3

2.4.4.4 Equipment Predictive Maintenance – PH4

Attualmente i dati grezzi provenienti dai sensori installati sui macchinari non vengono usati in alcun modo, e il monitoraggio si basa soltanto su *warning* ed errori. Per poter fare un passo in più, la richiesta è quella di integrare il risultato delle fasi precedenti, in particolare delle prime due, per identificare le anomalie basandosi sui dati inviati direttamente dai sensori e fornire, quindi, un supporto per la previsione di allarmi bloccanti.

L'utente utilizzatore si troverà ad interagire con una tabella con aggiornamento quasi real-time, che mostri, per ogni timestamp, quali *warning* sono stati rilevati dall'algoritmo di *Anomaly Detection* (Figura 2.13).

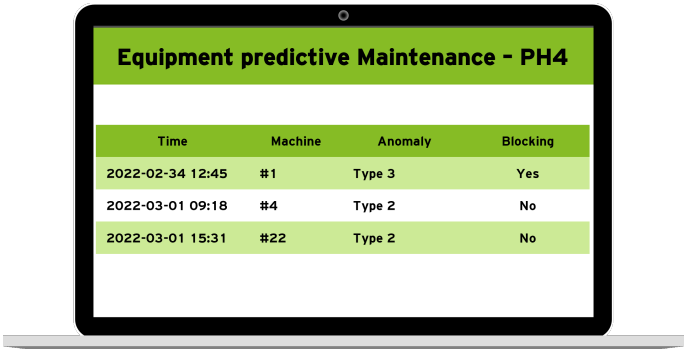

Figura 2.13: Mock-up relativo a Equipment Predictive Maintenance - PH4

2.4.4.5 Non-Conformity Correlation Analysis

Al momento l'attività di ricerca dei veicoli non conformi allo standard di produzione è eseguita manualmente dal personale di Quality, attraverso delle analisi condotte a valle della fase produttiva.

Nell'ottica di ridurre le attività manuali e di aumentare l'oggettività e l'allineamento dei controlli, si vuole definire un nuovo strumento di analisi di dati in grado di identificare pattern e correlazioni tra le non-conformità del veicolo. L'obiettivo è quello di supportare il team di Quality nel risalire la linea produttiva, in modo da facilitare l'identificazione dello step nel quale è stata introdotta la non-conformità, potendo, quindi, agire tempestivamente, e non solo al termine del processo produttivo.

2.4.4.6 Material Flow Optimization

La richiesta, in questo caso, è di un supporto per la gestione dei flussi di materiale dal magazzino verso la linea produttiva (Figura 2.14). Nello specifico si vogliono ottimizzare lo spazio nel magazzino e le tempistiche di movimentazione delle risorse verso i macchinari. Assicurando un corretto e immediato stoccaggio delle forniture, sarà anche possibile automatizzare il riordino dei materiali per i quali si stanno esaurendo le disponibilità. Ottimizzando tali operazioni, si ridurrà, di conseguenza, anche il rischio di bloccare le vetture sulla linea per la mancata disponibilità di materiale.

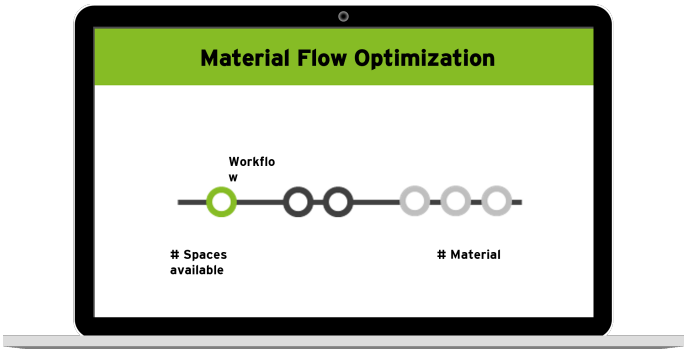

Figura 2.14: Mock-up relativo alla Material Flow Optimization

2.4.4.7 Supply Chain Control Tower

Attualmente il monitoraggio degli ordini richiede il passaggio attraverso diversi strumenti, ognuno dei quali contenente una base di dati diversa. La richiesta è quella di semplificare l'operazione e consentire il monitoraggio *Real-Time* degli ordini attraverso uno strumento unico, che sia, quindi, un punto raccolta dei dati riguardanti i fornitori, i trasporti e i magazzini.

L'outcome atteso è una *dashboard* che consenta di tracciare completamente l'ordine, dal fornitore, attraverso il trasporto, fino allo stato nel magazzino (Figura 2.15). Analizzando tale *dashboard* sarebbe, poi, possibile determinare le tempistiche medie per le operazioni di logistica ed identificare eventuali colli di bottiglia per poterli gestire, con la possibilità di integrare, in un futuro prossimo, un algoritmo che stimi la probabilità di ritardo dell'ordine.

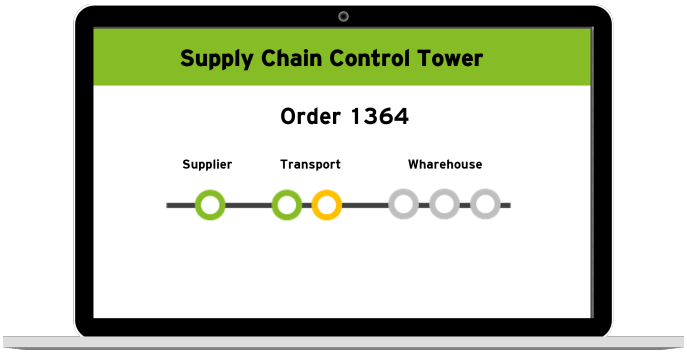

Figura 2.15: Mock-up relativo alla Supply Chain Control Tower

2.4.5 Cross-Business-Units

2.4.5.1 GeoMarketing Model – Product Marketing & After Sales

A livello strategico, la decisione riguardante l'apertura di nuove concessionarie è attualmente presa sulla base di analisi condotte manualmente da un'analista. Quest'ultimo associa ad ogni veicolo un'area geografica di appartenenza, determinata dall'indirizzo di residenza del proprietario, dichiarato in fase di acquisto.

Per poter avere una base più solida e aggiornata sulla quale prendere questo tipo di decisioni di business, l'azienda vuole avere a disposizione una mappatura geografica di tutte le vetture a livello globale, con l'obiettivo di identificare la reale regione di appartenenza e la distanza dal concessionario più vicino (Figura 2.16). In questo modo sarebbe possibile simulare l'impatto di apertura di un nuovo *dealer*, ma anche *clusterizzare* i clienti sulla base dei luoghi frequentati, per poter poi proporre prodotti o servizi aggiuntivi, più in linea con gli interessi evidenziati da ogni *cluster*.

Figura 2.16: Mock-up relativo al GeoMarketing

2.4.5.2 Predictive Maintenance – Strategy & Connectivity, Product Marketing e After Sales

Gli ultimi modelli di auto rilasciati in produzione da qualche anno a questa parte montano a bordo una serie aggiuntiva di sensori, che inviano dati alla casa madre con una frequenza di campionamento di 0,1 Hz. Su questi modelli si vogliono definire degli algoritmi di *Predictive Maintenance*, in grado di prevedere quando potrebbe verificarsi un possibile guasto, in modo tale da poter preventivamente adoperarsi, con opportuni interventi di manutenzione, al fine di evitare che il *fault* si traduca in un *failure* del veicolo.

Questi algoritmi consentirebbero non solo di aumentare l'affidabilità dei veicoli e ridurre i costi degli interventi di garanzia, ma anche di offrire un servizio migliore al cliente; il sistema, infatti, permetterebbe ai *dealer* di essere proattivi nel contattare loro stessi la clientela, a valle di una segnalazione di necessità di intervento ricevuta dal sistema.

In questo caso l'azienda desidera avere a propria disposizione un *dashboard*, filtrabile per vettura, che evidenzi eventuali guasti nella centralina del veicolo e, sulla base delle previsioni effettuate, fornisca una stima dello stato di usura dei componenti e di quando questi potrebbero ricadere in uno stato di *fault* (Figura 2.17).

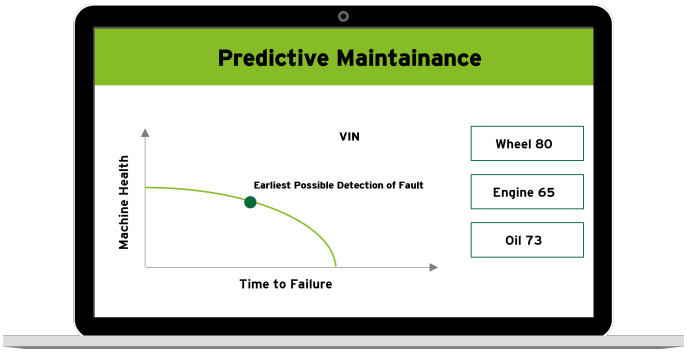

Figura 2.17: Mock-up relativo alla Predictive Maintenance

2.4.5.3 Personalized Customer Experience

Per gli stessi modelli per i quali si intendono definire algoritmi di *Predictive Maintenance* (Sezione 2.4.5.2), si vogliono anche sfruttare i dati dei sensori per determinare la tipologia di ambiente dove il cliente utilizza la vettura (città, pista, montagna, etc). Integrando questa informazione con la conoscenza dei servizi attivi di cui il cliente fa più uso, sarà possibile offrire *optional* e servizi integrativi *customizzati*, che tengano conto di questi aspetti e consentano, quindi, di garantire una esperienza d'uso personalizzata, senza la necessità di disturbare gli utenti con dei questionari telefonici o telematici.

Il *desiderata* è un *report* che, per ogni cluster, evidenzi le preferenze rilevate dalle analisi condotte (Figura 2.18).

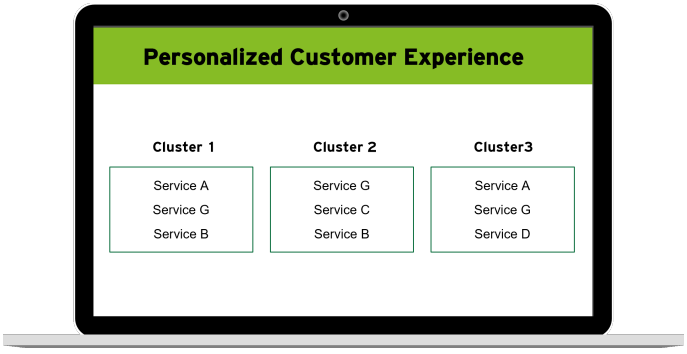

Figura 2.18: Mock-up relativo alla Personalized Customer Experience

2.5 Prioritizzazione degli Use Case

Per decidere quali Use Case sviluppare prima, è stato definito uno schema che ne evidenzia le interdipendenze (Figura 2.19).

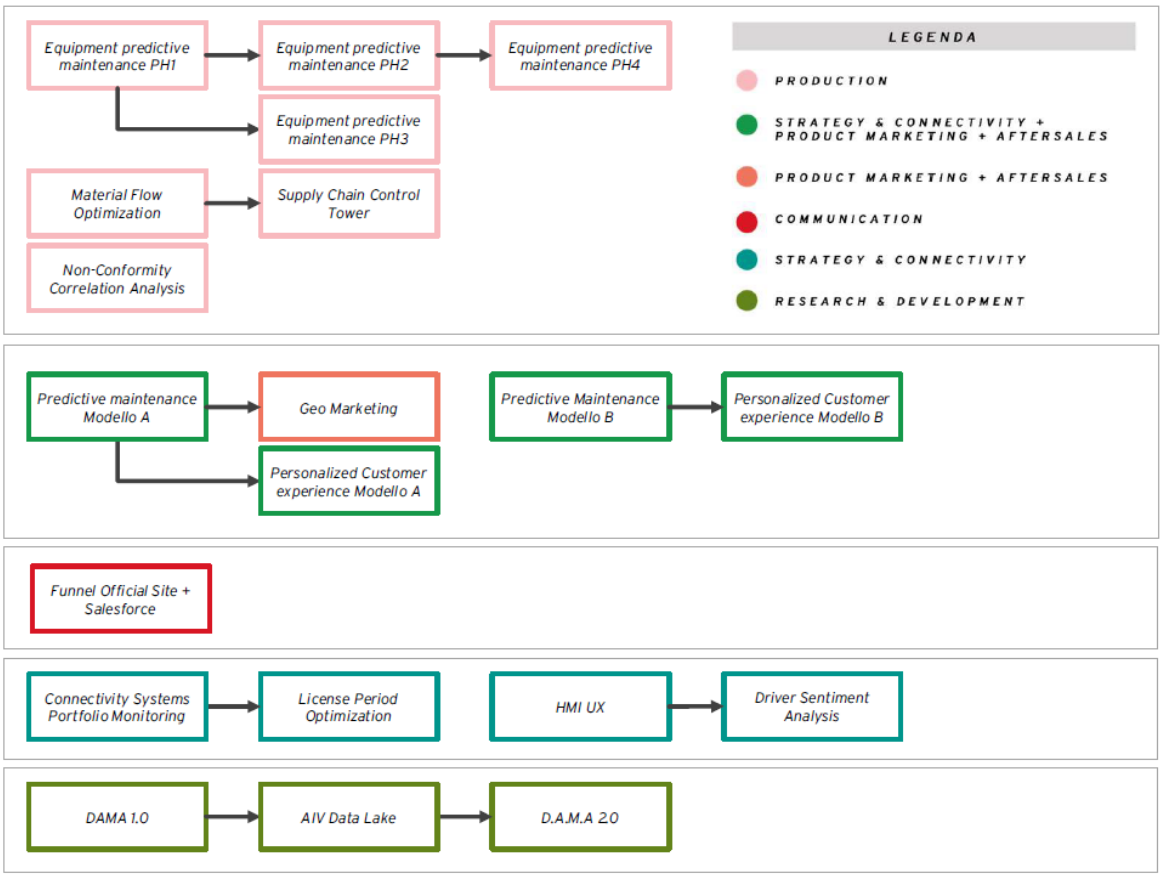

Figura 2.19: Schema delle dipendenze degli Use Case

Come secondo step, avvalendosi anche di questo schema, si è definito un set di criteri per valutare i casi d'uso e attribuire loro dei punteggi rispetto a:

- impatto sulla produzione;
- impatto sul cliente;
- impatto sui processi;
- impatto sull'azienda;
- possibilità di abilitare altri Use Case;
- disponibilità dei dati;
- restrizioni dovute alle norme sulla privacy;
- architettura tecnologica necessaria;
- dipendenza da altri Use Case;
- maturità dello Use Case.

La sommatoria pesata di questi valori ha permesso di ottenere un punteggio finale per ogni caso d'uso. I risultati dei pesi di ogni Use Case sono stati razionalizzati e riassunti nella matrice di impatto-fattibilità; la Figura 2.20 li riporta già ordinati per valore di prioritizzazione decrescente.

Overview degli Use Case	Impatto sul Business					Fattibilità				Maturità	Prioritizzazione
Nome dello Use Case	Produzione	Cliente	Processi	Azienda	Abilitatore altri UC	Disponibilità Dati	Restizioni Policy	Architettura	Dipendenza da altri UC	Probabilità di successo	Punteggio Finale
Predictive Maintenance - Model A											2
Equipment Predictive Maintenance - PH1											1,81
GeoMarketing Model											1,71
Personalized Customer Experience - Model A											1,68
Material Flow Optimization											1,64
AIV Data Lake											1,6
Equipment Predictive Maintenance - PH2											1,6
Equipment Predictive Maintenance - PH3											1,55
Funnel Official Site + Salesforce											1,55
HMI UX											1,49
Connectivity systems portfolio monitoring											1,48
D.A.M.A. (Digital Asset Management Assistant)											1,46
Supply Chain Control Tower											1,45
Predictive Maintenance - Model B											1,45
Non-Conformity Correlation Analysis											1,42
Equipment Predictive Maintenance - PH4											1,27
Personalized Customer Experience - Model B											1,26
License Period Optimization											0,93
Driver Sentiment Analysis											0,92
D.A.M.A. (Digital Asset Management Assistant) 2.0											0,77

Figura 2.20: Prioritizzazione degli Use Case: matrice impatto-fattibilità

Da queste valutazioni si evince quindi che il primo Use Case da sviluppare è quello di *Predictive Maintenance* sul modello d'auto di tipo A. Pertanto, da qui in avanti, la trattazione verterà soltanto su tale argomento, forti del fatto che, anche dal punto di vista della definizione del Data Lake aziendale, l'agilità offerta dalla scelta di implementarlo in versione cloud consentirà la successiva integrazione di tutti gli ulteriori servizi necessari a soddisfare gli altri Use Case.

CAPITOLO 3

Infrastruttura Cloud

Da questo capitolo in poi la trattazione si occuperà unicamente dello Use Case di Predictive Maintenance sul Modello A, il cui sviluppo, come evidenziato dalla prioritization matrix in Figura 2.20, risulta essere prioritario rispetto a quello di tutti gli altri. Dopo una breve introduzione relativa alla piattaforma cloud che si è deciso di utilizzare per l'implementazione del Data Lake, verrà presentata l'architettura definita e tutti i suoi componenti, specificandone l'utilizzo fatto in questo caso specifico.

3.1 Premessa

Come già anticipato in 2.2, la casa automobilista ha deciso di implementare il Data Lake in versione cloud. Per quanto concerne il provider invece, la scelta è stata quella di avvalersi dei servizi messi a disposizione da Amazon Web Services, leader[1] provider nel campo, che ad oggi divide il mercato con i suoi due principali competitor: Microsoft Azure Cloud e Google Cloud Services (Figura 3.1).

3.2 Amazon Web Services (AWS)

È la piattaforma cloud di proprietà del gruppo Amazon. Lanciata nel 2002, ad oggi copre il 59% circa del fatturato di Amazon, facendosi strada come leader nel settore, con crescita rispetto all'anno precedente di oltre il 32%.

Dati alla mano, AWS risulta essere la piattaforma cloud più completa e utilizzata al mondo che include offerte di Infrastructure-as-a-Service (IaaS) e Platform-as-a-Service (PaaS). I servizi AWS sono oltre 200 e offrono soluzioni scalabili, flessibili e affidabili, per il calcolo, lo stoccaggio, i database, l'analisi e altro ancora; disponibili in quasi tutte le regioni del mondo. L'azienda vanta al suo attivo milioni di clienti, tra i quali spiccano varie agenzie governative statunitensi e altri come, ad esempio, Adobe, Netflix e iRobot.

3.3 Architettura Cloud del Data Lake

La scelta di utilizzare AWS è dettata da una serie di fattori, tra i quali il fatto che offre il portfolio di servizi più sicuro, scalabile, completo ed economicamente vantaggioso, disponibile momentaneamente sul mercato e che consente ai clienti di costruire Data Lake nel cloud

[1] `https://www.gartner.com/doc/reprints?id=1-271OE4VR&ct=210802&st=sb`

Figura 3.1: Magic Quadrant Gratner per l'infrastruttura cloud e i servizi di piattaforma, edizione 2021

e analizzarne i dati, inclusi quelli provenienti dai dispositivi IoT, con una varietà di approcci analitici che includono svariate tecniche di *Machine Learning*.

Lo studio e la valutazione dei servizi disponibili su AWS, ha consentito di conoscerne le caratteristiche e funzionalità principali. Sulla base di queste conoscenze, si è poi stati in grado di delineare l'architettura cloud del Data Lake, riportata in Figura 3.2. Nelle sezioni seguenti verranno presentati gli elementi che la compongono, spiegando le caratteristiche dei servizi e il loro utilizzo, allo scopo di giustificarne l'adozione.

3.3.1 Amazon Simple Storage Service - S3

Amazon Simple Storage Service, comunemente chiamato Amazon S3, è un servizio di *storage* che offre scalabilità, disponibilità dei dati, sicurezza e ottime prestazioni. È di facile utilizzo, gestisce facilmente i dati su qualsiasi scala, consentendo inoltre di configurare controlli di accesso, mirati a soddisfare specifici requisiti aziendali, organizzativi e di conformità.

S3 è un servizio di *storage* ad oggetti; questo significa che ogni dato, che può avere dimensione fino a 5 TB, viene memorizzato come oggetto all'interno di alcune risorse, che prendono il nome di *bucket*. L'oggetto è composto dal file e dai metadati che lo descrivono ed è identificato univocamente da una chiave; il *bucket* è un contenitore di oggetti, che al suo interno possono essere organizzati in *folder*, che vengono comunque considerati da S3 come degli oggetti.

Per caricare i dati in Amazon S3 è necessario come prima cosa creare un *bucket*, specificandone nome e regione AWS; solo a questo punto sarà consentito il caricamento dei file. Alla creazione *bucket* e oggetti al loro interno, sono privati e inaccessibili perfino al creatore; l'accesso è infatti regolato da permessi che devono essere esplicitamente concessi tramite

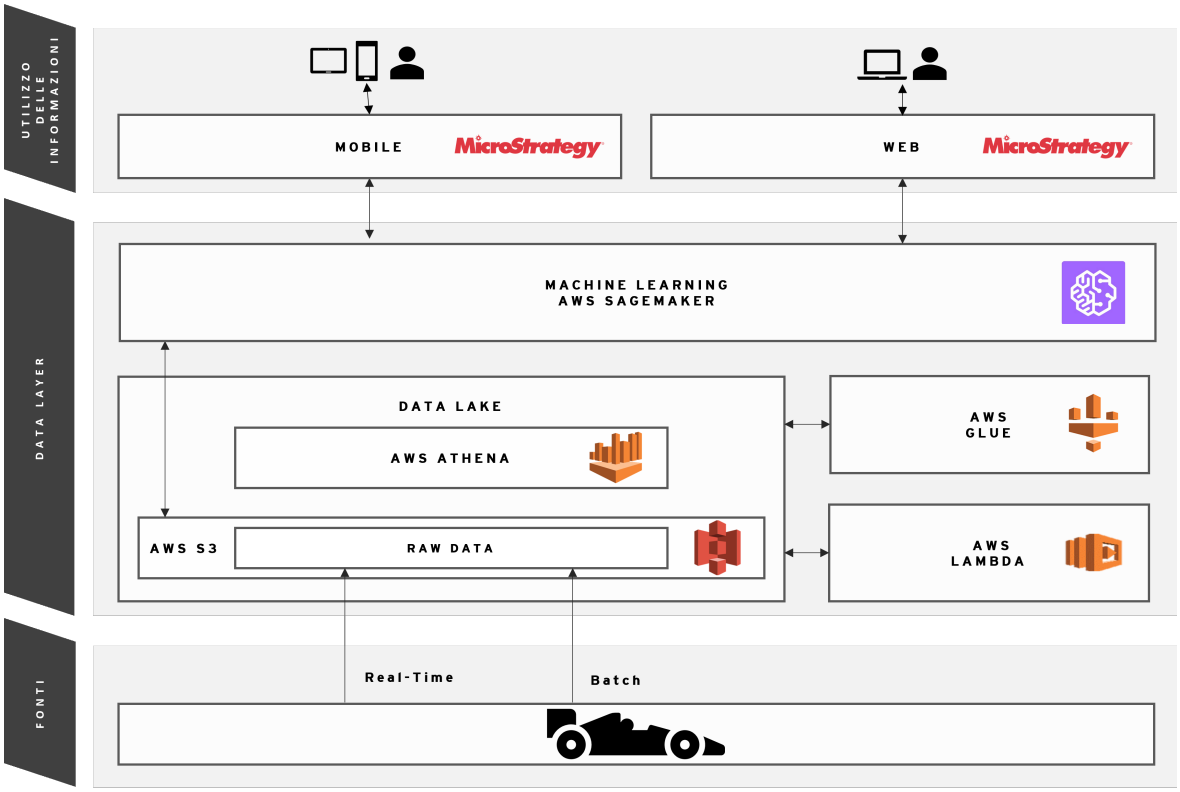

Figura 3.2: Architettura Cloud del Data Lake progettata per lo Use Case di Predictive Maintenance sul Modello A

AWS Identity and Access Management (IAM), un servizio che gestisce il controllo granulare degli accessi per tutti i servizi della piattaforma.

All'interno dell'architettura della Figura 3.2 progettata per lo Use Case di *Predictive Maintenance*, S3 è il servizio scelto per fungere da vero e proprio Data Lake; al suo interno è stato definito il *bucket "datalake-connectivity"* che raccoglie il risultato della fase di Data Ingestion, ovvero i dati grezzi provenienti, real-time e in batch, dai veicoli sensorizzati. All'interno di S3 verranno archiviati anche i dati risultanti dalle attività di Data Preparation, affinché possano poi essere accessibili e utilizzabili con tutti i restanti servizi.

3.3.2 AWS Lambda

Lambda è un servizio *serverless* che esegue il codice su un'infrastruttura di calcolo ad alta disponibilità e gestisce tutta l'amministrazione delle risorse di elaborazione, compresa la manutenzione del server e del sistema operativo, nonché il monitoraggio delle esecuzioni e i *log* del codice. Il servizio è in grado di scalare automaticamente, passando dalla gestione di poche richieste al giorno, a migliaia al secondo; chiaramente viene fatturato solo il tempo di calcolo.

Consente di eseguire il codice praticamente per qualsiasi tipo di applicazione o servizio back-end. Tutto quello che occorre fare, è caricare il proprio codice in uno dei linguaggi supportati; questo verrà organizzato in funzioni Lambda che saranno eseguite solo quando richiamate tramite l'apposita API , oppure in risposta agli eventi generati da altri servizi AWS.

Nell'architettura progettata da Deloitte, la ricezione di nuovi dati fungerà da trigger per una AWS Lambda che a sua volta attiverà i processi successivi, ovvero dei job definiti

all'interno di AWS Glue (Sottosezione 3.3.3).

3.3.3 AWS Glue

AWS Glue è un servizio *serverless* per fare ETL (*Extract, Transform, Load*) dei dati, che rende semplice e facile tipizzarli, pulirli, arricchirli e spostarli, in modo affidabile e sicuro, attraverso i vari i *bucket* S3 e flussi.

È costituito da un repository di metadati centralizzato, noto come AWS Glue Data Catalog, un motore ETL che genera automaticamente codice Python o Scala, e uno scheduler flessibile che gestisce la risoluzione delle dipendenze e il monitoraggio dei processi.

All'interno di Glue sono disponibili vari strumenti per la gestione dei dati e dei processi; qui di seguito verranno presentati solo quelli utilizzati specificandone il modo di impiego:

- *Crawler*: si connette a una sorgente dati, S3 nel caso in analisi, e, tramite una serie di classificatori, inferisce autonomamente lo schema, per poi creare una tabella nell'AWS Glue Data Catalog, rendendola disponibile alle successive interrogazioni SQL eseguibili tramite Amazon Athena. Chiaramente, è anche possibile apportare manualmente modifiche allo schema, nel caso sia necessario.

- *Glue Studio*: è una interfaccia grafica che rende facile creare, eseguire e monitorare i job di ETL in AWS Glue, componendoli tramite una serie di blocchetti grafici che rappresentano la sorgente, il target e tutte le elaborazioni intermedie. Creata la pipeline di ETL, questa viene automaticamente tradotta in uno script Apache Spark che verrà eseguito sulla *engine* di Glue; in alternativa è anche possibile definire il job direttamente tramite uno script Python o Apache Spark.

  Per lo Use Case in questione, sono stati definiti una serie di job, alcuni direttamente tramite script Python e altri in modalità grafica. La loro esecuzione, in alcuni casi è affidata ad una Lambda function, *triggerata* dal caricamento di nuovi dati, in altri casi, invece, a una schedulazione temporale. In aggiunta a questi, ci sono anche dei job che si attivano vicendevolmente, oppure sono eseguiti automaticamente a seguito del caricamento di nuovi oggetti nel *bucket* dedicato al Data Lake. Per i job che prevedono un output, il target predefinito è sempre il *bucket "datalake-connectivity"* di S3.

3.3.4 AWS Athena

Amazon Athena è un servizio di *query* interattivo che semplifica l'analisi dei dati in Amazon S3 con espressioni SQL standard. Anche questo è un servizio *serverless*, quindi non c'è nessuna infrastruttura da impostare o gestire, e si paga solo per le *query* che si eseguono. È ottimo anche dal punto di vista delle prestazioni; infatti scala automaticamente, eseguendo le *query* in parallelo, garantendo quindi risultati rapidi anche con grandi quantità di dati e *query* complesse.

Athena offre l'integrazione preconfigurata con AWS Glue Data Catalog, consentendo di interrogare le tabelle definite al suo interno e i cui dati sottostanti si trovano in S3.

3.3.5 AWS SageMaker

Amazon SageMaker è un servizio di *Machine Learning* completamente gestito, con il quale è possibile costruire e addestrare, rapidamente e facilmente, modelli di *Machine Learning* per poi distribuirli direttamente in un ambiente pronto per essere rilasciato in produzione.

Mette a disposizione dell'utente i principali algoritmi di apprendimento automatico, utilizzabili per diverse tipologie di analisi e dati, già ottimizzati per l'esecuzione efficiente su grandi quantità di dati locati in un ambiente distribuito; oltre a questi è chiaramente possibile

anche definire dei propri algoritmi e *framework*; in questo caso sarà poi il servizio ad adattarsi ai flussi di lavoro specifici.

Qui di seguito verranno riportati soltanto i componenti di AWS SageMaker utilizzati per lo sviluppo dello Use Case:

- *SageMaker Studio*: un ambiente integrato di *Machine Learning* dove analizzare i dati, costruire, addestrare, distribuire i modelli, tutto in un'unica applicazione
- *SageMaker Studio Lab*: un servizio gratuito che dà accesso alle risorse di calcolo AWS in un ambiente basato su JupyterLab

3.3.6 Piattaforma BI: MicroStrategy

Dal lato della Business Intelligence, l'analisi condotta sarà fornita attraverso la MicroStrategy Analytics Platform, strumento di BI scelto dal cliente e già utilizzato per le altre attività del team, sebbene esso non sia tra i migliori disponibili sul mercato (Figura 3.3).

Figura 3.3: Magic Quadrant Gratner per Analytics e Business Intelligence, edizione 2022

La piattaforma permette di implementare diversi tipi di visualizzazioni e di presentare il risultato dei processi analitici attraverso *Report*, *Documenti* o *Dossier*.

Qui di seguito verranno presentati i componenti di MicroStrategy inseriti all'interno della architettura cloud progetta:

- *MicroStrategy Web*: permette agli utenti di eseguire tutti i principali tipi di Business Intelligence (BI), progettare e creare *Report*, Documenti e *Dossier* in modo autonomo, attraverso un browser web.
- *MicroStrategy Mobile*: permette agli utenti di sfruttare la potenza analitica di MicroStrategy per eseguire *Report*, Documenti e *Dossier* direttamente dai loro dispositivi mobili Apple iOS e Android, in modalità Real-Time.

CAPITOLO 4

Dataset Strategy & Connectivity

Lo Use Case in analisi si occupa di manutenzione predittiva su uno specifico modello della gamma prodotta dalla casa automobilistica. Il modello in questione è uscito solo pochi anni fa, quando già il cliente iniziava a valutare la possibilità di introdurre questa tipologia di manutenzione sui propri veicoli. È proprio per questo motivo, quindi, che tutti i veicoli su strada di questo modello sono già dotati di una serie di sensori aggiuntivi rispetto allo standard, montati ad hoc proprio per l'acquisizione di variabili che la casa automobilistica, a suo tempo, ha ritenuto potessero essere utili per la definizione di un modello in grado di prevedere il verificarsi di possibili guasti.

Entrando sempre di più nel dettaglio del progetto di Predictive Maintenance, il presente capitolo intende, quindi, presentare il dataset di lavoro. Verranno forniti dettagli in merito a come si compone il dataset, sia dal punto di vista dei veicoli che delle informazioni acquisite, evidenziando alcune problematiche che lo affliggono.

4.1 Variabili acquisite

La casa produttrice ha fornito a Deloitte i dati acquisiti, da Gennaio 2021 a Febbraio 2022, dai sensori montati su 1457 veicoli del modello sul quale si vuole eseguire manutenzione predittiva, per un totale di oltre 48 milioni di record. Ognuno di questi contiene il vin, ovvero il numero di telaio che identifica univocamente ogni veicolo nel mondo, l'istante di acquisizione dei dati, quello di scrittura del record sul Data Lake, oltre ai valori letti da 78 sensori che forniscono informazioni su diverse componenti del veicolo: dal motore alle gomme, dal cambio allo sterzo, dalle portiere alla batteria, etc.

I veicoli presentano delle strutture meccatroniche molto complesse; si compongono, infatti, di una serie di sottosistemi, che comprendono processi elettromeccanici, attuatori e sensori. Pertanto, per la corretta valutazione e interpretazione dei dati, sarebbe stato di fondamentale importanza acquisire conoscenza di dominio e, in particolar modo, comprendere il significato di ogni variabile. Nonostante si siano tenute numerose riunioni con diverse divisioni e rappresentanti dell'azienda cliente, ad oggi non si è stati in grado di identificare una figura capace di associare un significato ad ogni variabile acquisita, né tanto meno un *range* di funzionamento nominale.

La Tabella 4.1 elenca solo alcune delle variabili in questione, corredate di significato fisico, nel caso in cui questo sia noto, di una semplice traduzione, in caso contrario. Purtroppo, talvolta, le informazioni fornite coincidono con quelle di altre variabili, dalle quali però differiscono di valore, rendendo ancora più complesso comprendere il reale significato del dato.

Informazione	**Descrizione**
vin	Numero di telaio, identificativo univoco del veicolo
created at	Timestamp di scrittura del record
speed	Velocità del veicolo
acc longitudinal	Accelerazione del veicolo
acc lateral	Accelerazione centrifuga del veicolo
driving mode	Modalità di guida inserita tra Corsa, Strada e Sport
slip angle	Imbardata (angolo di slittamento)
slip angle quality bit	Flag di slip angle
slip direction	Direzione di slittamento
rws angle	Angolo RWS o movimento in mm dell'attuatore
rws angle quality bit	Flag di rws angle
rws sign	Segno di RWS
wheel ang vel fl	Velocità angolare della ruota anteriore sinistra
wheel ang vel fr	Velocità angolare della ruota anteriore destra
wheel ang vel rl	Velocità angolare della ruota posteriore sinistra
wheel ang vel rr	Velocità angolare della ruota posteriore destra
esp status	Stato del sistema di controllo dinamico della stabilità
start stop info	Informazioni di start e stop
gearbox status	Stato del cambio
activ launch control	Indica se è attivo il launch control che consente una partenza scattante, tipica dei veicoli sportivi
drive mode status	Stato della modalità di guida
driving mode rws	Modalità di guida rws
km tachometer	Km sul tachimetro
drive mode for eps	Modalità di guida nella si è attivato il controllo dinamico della stabilità
battery level p	Livello della batteria in percentuale
throttle	Pressione sul pedale dell'acceleratore
throttle quality bit	Flag su throttle
brake pressure	Pressione freni
brake pressure quality bit	Flag su brake pressure
angle	Angolo di sterzata
direction	Valore assoluto dell'angolo di sterzata
direction quality bit	Flag di direction
angular speed	Velocità angolare di sterzata
angular direction	Angolo di sterzata
wheel position	Posizione del volante

torque	Coppia di sterzo trasmessa dal conducente al volante
torque quality bit	Flag su torque
torque sign	Segno della coppia di sterzo trasmessa dal conducente al volante
engaged auto	Marcia inserita quando si usa il cambio automatico
engaged manual	Marcia inserita quando si usa il cambio manuale
torque instant	Coppia istantanea fornita dal motore
power	Potenza
power instant	Potenza istantanea generata dal motore
rpm	Giri del motore al minuto
rpm quality bit	Flag di rpm
coolant temperature	Temperatura del liquido di raffreddamento del motore
coolant temperature quality bit	Flag di coolant temperature
oil temperature	Temperatura dell'olio
oil temperature quality bit	Flag di oil temperature
fuel level	Contenuto del serbatoio del carburante in litri
latitude	Latitudine in cui si trova il veicolo
longitude	Longitudine in cui si trova il veicolo
altitude	Altitudine in cui si trova il veicolo
units	Flag che indica se l'unità di misura è in kmh o mph
4 wheel drive error	Flag che indica se la trazione integrale è in errore
4 wheel drive e msg	Messaggio di errore per la trazione integrale
coupling overheat	Indica se il giunto si è surriscaldato ed è modalità di protezione
4 wheel drive oil temp coupling	Temperatura dell'olio nel giunto a 4 ruote motrici
vitality	Chiave inserita
battery level	Livello della batteria
bonnet	Cofano aperto/chiuso
door driver lock	Porta del conducente bloccata/sbloccata
door driver open	Porta del conducente aperta/chiusa
door passenger locked	Porta del passeggero bloccata/sbloccata
door passenger open	Porta del passeggero aperta/chiusa
fuel tank	Serbatoio del carburante
fuel tank n a	Serbatoio del carburante
time	Timestamp che indica l'istante di acquisizione dei campioni
window driver perc open	Finestrino del conducente aperto/chiuso
window passenger perc open	Finestrino del passeggero aperto/chiuso
trunk lid	Portabagagli aperto/chiuso

hand brake	Freno a mano tirato/non tirato
roof open	Tettino panoramico aperto
roof closed	Tettino panoramico chiuso
roof locked	Tettino panoramico bloccato
roof intermediate	Tettino panoramico semiaperto
satellites number	Numero di satelliti
horizontal accuracy	Accuratezza orizzontale
vertical accuracy	Accuratezza verticale
position dop	Diluizione della precisione della stima della posizione
fix status	Tipo di tecnica utilizzata per determinare la posizione del veicolo

Tabella 4.1: Descrizione delle informazioni contenute nel dataset di Connectivity

4.2 Approfondimento sui vin del dataset

I vin contenuti nel dataset sono relativi a veicoli dislocati nei diversi paesi del mondo; alcuni di questi sono di proprietà dei clienti della casa automobilistica; altri, invece, sono veicoli di test, posseduti della casa madre e messi su strada per essere sottoposti a degli stress test. Queste tipologie di test vengono condotte in luoghi del mondo che si trovano in condizioni climatiche molto diverse tra loro, dalle più impervie, caratterizzate da caldo torrido o freddo artico, a quelle più miti. Un altro fattore che varia da test a test è la quota; l'altitudine, infatti, può influire sulle prestazioni del veicolo, e questi test servono all'azienda cliente a studiare se, come e quanto, questo influisca sul corretto funzionamento dell'auto.

Inoltre, non bisogna dimenticarsi che le macchine sportive possono essere guidate come comuni auto, ma possono anche essere portate in pista, dove sono libere di mostrare la loro vera natura. I test in pista consentono di spingere al massimo la vettura e valutare una serie di fattori come, ad esempio, il modo in cui si modifica l'usura dei componenti rispetto ad una guida più standard, condotta all'interno di un contesto urbano.

Complessivamente, quindi, possiamo dire che questa tipologia di test serve a valutare il veicolo in diverse condizioni e modalità di utilizzo, verificandone la qualità ed evidenziandone eventuali problematiche, sanabili con aggiornamenti software della centralina o di cui tener conto nella progettazione della nuova versione del modello. A questo gruppo di veicoli, infatti, appartengono anche alcuni prototipi delle nuove versioni di questo modello di auto, che potrebbero, quindi, presentare persino delle differenze costruttive rispetto allo standard attualmente definito in produzione. Non avendo a nostra disposizione un modo per distinguere i veicoli sulla base delle loro caratteristiche costruttive, si è ritenuto inopportuno coinvolgere i veicoli di test nelle analisi. Utilizzare tutti i vin a disposizione, senza, quindi, isolare i 97 di test, potrebbe, infatti, portare a definire un modello inesatto, o quantomeno influenzato da dati prodotti da sessioni di guida volutamente inusuali o, perfino, da caratteristiche costruttive diverse. È proprio per questo motivo, quindi, che nelle prime fasi di elaborazione del dataset, tutti i record associati ai vin di test verranno elisi.

CAPITOLO 5

Panoramica sugli approcci di Machine Learning valutati

Attualmente, il Machine Learning è utilizzato in molti aspetti della vita quotidiana; ci veniamo a contatto, senza neanche accorgercene, quando facciamo acquisiti online, usiamo i social media, facciamo operazioni bancarie, etc. Tre fattori possono essere menzionati come i motori dello sviluppo sorprendente del Machine Learning: la disponibilità dei dati, le scoperte nello sviluppo degli algoritmi e i progressi nella potenza di calcolo. Il tipo di metodo e di algoritmo di Machine Learning da utilizzare, però, è dettato unicamente dall'applicazione e dai dati disponibili.

In questo capitolo verranno presentate le famiglie di tecniche di Machine Learning valutate per lo sviluppo dello Use Case di Predictive Maintenance. Per ognuna di esse verrà fornita una breve descrizione, utile a giustificare se e perché quella tipologia di approccio sia stata applicata o meno.

5.1 Apprendimento Supervisionato

L'apprendimento supervisionato viene eseguito utilizzando una verità di base, cioè una conoscenza preliminare di quali dovrebbero essere i valori di uscita predetti dal modello.

La definizione del modello consiste nel fornire in input all'algoritmo di *Supervised Learning* un insieme di dati per i quali si conosce già l'output desiderato, tecnicamente chiamato *ground-truth*. Il cuore del processo è l'addestramento, ovvero la fase durante la quale l'algoritmo apprende autonomamente dai dati come generare un output, che sia il più simile possibile al *ground-truth*. Il rischio, in questo caso, è quello di ricadere in *overfitting*, ovvero ottenere un modello che, sebbene sembri operare correttamente considerando gli elevati punteggi ottenuti rispetto alle metriche di valutazione scelte, in realtà, è un modello pressoché inutilizzabile. Le ottime performance, infatti, non sono dovute al fatto che il modello ha appreso correttamente il modo in cui prevedere l'output desiderato, ma sono piuttosto dovute ad un suo eccessivo adattamento ai dati training, che non consentirà poi, ad esso, di operare correttamente su dati nuovi, come quelli dell'insieme di test.

Alla famiglia degli algoritmi di *Supervised Learning* appartengono quelli di classificazione e regressione, entrambi utilizzabili per predire dei valori, che sono categorici, quindi classi, nel primo caso, numerici nel secondo.

Effettuare manutenzione predittiva significa stimare l'attuale stato di salute del veicolo e predire, utilizzando un modello appositamente addestrato, la Remaining Useful Life (RUL) del veicolo. La richiesta della casa automobilistica, però, non è quella di conoscere esattamente questo valore per ogni veicolo, quanto piuttosto di stimare, sulla base dei dati a disposizione, quanti veicoli avranno bisogno di interventi di manutenzione nel mese a venire. Questo non solo consentirebbe di allertare preventivamente i *dealer* di riferimento, ma anche di ridurre i

costi di garanzia e di porsi in maniera proattiva verso il cliente, richiamando la vettura prima che si verifichino i primi sintomi di guasto.

Il modello di *Machine Learning*, quindi, dovrà essere in grado di predire, a fronte di un *array* di valori ricevuto in ingresso, se nel mese successivo quel veicolo necessiterà di interventi manutentivi. La classe di algoritmi che si adatta perfettamente a questo compito è, chiaramente, quella dei classificatori binari; il problema è che, come spiegato in precedenza, l'apprendimento supervisionato ha necessità di disporre di un sottoinsieme di dati per i quali sia già noto, o comunque determinabile, l'output che il modello dovrebbe essere in grado di prevedere. Purtroppo, però, i dati forniti non sono già etichettati e, nonostante le numerose riunioni con il business, non è stata fornita alcuna conoscenza del mondo della meccanica dell'auto che potesse, in qualche, modo aiutare a costituire una base di partenza utile in questo senso. In aggiunta a ciò, non sono state fornite informazioni neanche relativamente alle soglie oltre le quali le variabili avrebbero descritto un comportamento indesiderato; pertanto, non è stato possibile definire un insieme di dati di training a causa dell'indeterminabilità del *ground-truth*.

Questo problema ha comportato, quindi, l'abbandono, quantomeno per il momento, dell'idea di applicare un apprendimento puramente supervisionato, ovvero quello dal quale ci si sarebbe aspettato di ottenere il miglior modello possibile, proprio grazie alla guida del *ground-truth*.

5.2 Apprendimento Rule-Based

Un sistema *Rule-Based* è un sistema intelligente in grado di trarre delle conclusioni, anche dette inferenze, a partire da una conoscenza di base, costituita da regole e fatti. Mentre i fatti si limitano ad asserire delle informazioni note nella *Knowledge Base*, le regole esprimono delle relazioni del tipo *if... then* e sono, quindi, espresse sotto forma di clausole composte da due parti: l'antecedente e il conseguente. Per popolare la *Knowledge Base* è, quindi, necessario codificare la conoscenza di un esperto umano nel dominio in analisi; è proprio per questo motivo che i sistemi *Rule-Based* sono anche conosciuti come sistemi esperti (*Expert Systems*).

Questo tipo di sistemi si basano su una programmazione dichiarativa, che differisce da quella procedurale, più comunemente utilizzata, proprio per il fatto che non viene in alcun modo dichiarato come utilizzare la base di conoscenza per arrivare alle conclusioni, quanto, piuttosto, le regole che devono essere rispettate dalle conclusioni. Ciò significa che i programmi dichiarativi hanno bisogno di un motore inferenziale, o ragionatore semantico, in grado di estrarre conoscenza sotto forma di conclusioni o conseguenze logiche, al termine del processo di inferenza.

Applicare un approccio di questo tipo per lo Use Case in questione avrebbe richiesto ancora di più, rispetto al *Supervised Learning*, la necessità di confrontarsi con un esperto di dominio; probabilmente la figura più idonea sarebbe stata quella di un meccanico, che avrebbe potuto fornire conoscenze fondamentali alla definizione della *Knowledge Base*. Anche in questo caso, però, non è stato possibile definire questo colloquio; pertanto, anche questa strada non è stata intrapresa.

5.3 Apprendimento Semi-Supervisionato

L'idea, in questo caso, è quella di valutare una strada intermedia tra l'approccio non supervisionato e quello supervisionato; in letteratura si fa anche riferimento a tale approccio con la definizione di supervisione debole. L'apprendimento semi-supervisionato, infatti, combina una piccola quantità di dati etichettati con una grande quantità di dati non etichettati.

I dati non etichettati, se usati insieme a una piccola quantità di dati etichettati, possono, infatti, portare ad un considerevole miglioramento nell'accuratezza dell'apprendimento.

L'apprendimento semi-supervisionato combina queste informazioni per superare i risultati che si potrebbero ottenere, scartando i dati non etichettati ed effettuando un apprendimento supervisionato, o scartando le etichette ed eseguendo un apprendimento non supervisionato. In sostanza, quindi, tramite l'applicazione dell'apprendimento induttivo ai dati etichettati, si comprende la logica che li lega alle *label* di *ground-truth* e, tramite apprendimento trasduttivo, la si applica ai restanti dati per assegnare ad essi una etichetta.

Complessivamente, quindi, lo sviluppo del modello si compone di due fasi, la prima è mirata alla previsione delle etichette per la parte di dataset che ne è sprovvisto; la seconda, di *fine-tuning*, è quella nella quale si cerca di migliorare il modello, fornendo in input ad esso tutti i dati a disposizione, con le relative etichette, originali e predette, al fine di diminuire l'errore e migliorarne l'accuratezza.

Come già spiegato, non avendo a disposizione le *label* di *grund-truth*, né una logica che consentisse di definirle, anche questa strada è stata momentaneamente accantonata.

5.4 Apprendimento Non Supervisionato

L'*Unsupervised Learning*, a differenza del *Supervised* e del *Semi-Supervised*, non ha bisogno di alcuna etichetta di *ground-truth*. Basandosi solo e unicamente sui dati a disposizione, questo tipo di apprendimento ha la capacità di estrarre degli *insight* del tutto sconosciuti, che potrebbero racchiudere informazioni di altissimo valore.

Per certi versi, si potrebbe dire che l'apprendimento non supervisionato ricalchi il modo di osservazione del mondo messo in atto dall'uomo, che quanto più osserva, tanto più apprende; allo stesso modo, il modello, quanti più dati processa, tanto più è in grado di migliorare le sue capacità, garantendo, via via, risultati sempre migliori e accurati.

Considerando che per questo Use Case l'unica conoscenza a disposizione risulta essere quella fornita dai dati stessi, appare chiaro che non ci sia altra possibilità se non quella di procedere con questa tipologia di approccio. A livello operativo, però, nessun algoritmo non supervisionato sarebbe stato sufficiente ad ottenere il risultato desiderato, ovvero prevedere la possibilità di guasto del veicolo. Pertanto, la scelta è stata quella di tentare un approccio diverso e più inusuale, ovvero l'approccio misto, presentato nella seguente sottosezione.

5.5 Approccio Misto: Non Supervisionato e Supervisionato

A differenza dell'approccio semi-supervisionato, che risulta essere una via di mezzo tra quello non supervisionato e quello supervisionato, la scelta fatta è stata quella di combinare a cascata le due tipologie, al fine di godere dei vantaggi di entrambe.

La decisione di intraprendere questa strada nasce principalmente da una necessità di progetto, dovuta al bisogno di sopperire alla mancanza di informazioni utili a definire una variabile target, che, come discusso nelle sezioni precedenti, avrebbe indubbiamente consentito di semplificare il lavoro con un più semplice e immediato modello di classificazione. Quello a cui si arriva è, però, un risultato di rilevante importanza.

Tendenzialmente, infatti, si è in grado di reperire le conoscenze di dominio necessarie, per cui questa tipologia di approccio viene utilizzato molto di rado. È proprio per questo motivo che il suo sviluppo assume un valore ancora più rilevante, non solo per il cliente, ma anche per la stessa Deloitte. All'interno di un'azienda di consulenza, infatti, è di fondamentale importanza la condivisione del *know-how* maturato grazie alle sfide che si è chiamati ad affrontare nei vari progetti. In questo caso specifico, una problematica di progetto si è tramutata nella possibilità di produrre e mettere a disposizione dell'intera azienda una *pipeline* da poter

applicare anche in altri contesti e per altri clienti, qualora si ripresentasse la problematica della quasi totale assenza di conoscenza di dominio necessaria ad operare.

L'idea a cui si è arrivati è quella di applicare un algoritmo non supervisionato per estrarre, direttamente dai dati, la conoscenza di cui non si è in possesso, al fine di poterla poi utilizzare per definire la variabile target dell'algoritmo supervisionato.

CAPITOLO 6

Sviluppo dello Use Case di Predictive Maintenance

Con questo capitolo si entra nel cuore dello sviluppo dello Use Case di Predictive Maintenance.

Per iniziare, verranno fornite alcune informazioni in merito alle scelte di sviluppo, relative al linguaggio utilizzato e al formato di salvataggio dei file prodotti dai vari processi.

Successivamente, verrà descritta l'elaborazione eseguita sul dataset, tralasciando le operazioni di pre-processing standard, a favore di quelle appositamente progettate per manipolare i dati a disposizione.

Infine, la Sezione "Pipeline di apprendimento automatico" si addentrerà, in via definitiva, nello sviluppo dell'approccio misto, anticipato nel precedente capitolo. Verranno forniti dettagli sugli algoritmi di apprendimento automatico, supervisionato e non, scelti, passando per lo sviluppo di metodi utili e per la definizione della logica di etichettatura dei dati.

6.1 Dettagli sulle scelte di sviluppo

6.1.1 PySpark

Lo sviluppo dell'intero Use Case è stato eseguito in AWS SageMaker, utilizzando Python; in particolare, dovendo effettuare elaborazioni che coinvolgono una grande mole di informazioni, è stata utilizzata la libreria PySpark, sia per il *pre-processing* dei dati, che per la definizione degli algoritmi.

PySpark è un'interfaccia per Apache Spark in Python, che supporta la maggior parte delle caratteristiche di Spark, come *Spark SQL*, *DataFrame*, *Streaming*, *MLlib* (*Machine Learning Library*) e *Spark Core*. Spark è un *framework* di elaborazione distribuita, particolarmente adatto ad essere utilizzato per elaborazioni dei Big Data, in quanto è in grado di elaborare grandi quantità di dati con una velocità che supera di oltre 100 volte quella di Pandas, una libreria Python molto utilizzata per il *pre-processing* dei dati.

Nella Figura 6.1 è riportato lo Spark Stack sul quale si basa anche PySpark.

Di seguito verrà fornita una descrizione soltanto per le componenti utilizzate nelle elaborazioni:

- *PySpark SQL*: è un modulo Spark per l'elaborazione di dati strutturati tramite linguaggio SQL. Il risultato viene restituito sotto forma di DataFrame, una struttura dati organizzata in righe e colonne, concettualmente equivalente a una tabella di una database relazionale.

- *MLlib*: è una libreria di apprendimento automatico, scalabile, che fornisce un insieme di strumenti che consentono di mettere a punto pipeline di apprendimento automatico, del tutto simili a quelle definibili con *scikit-learn*.

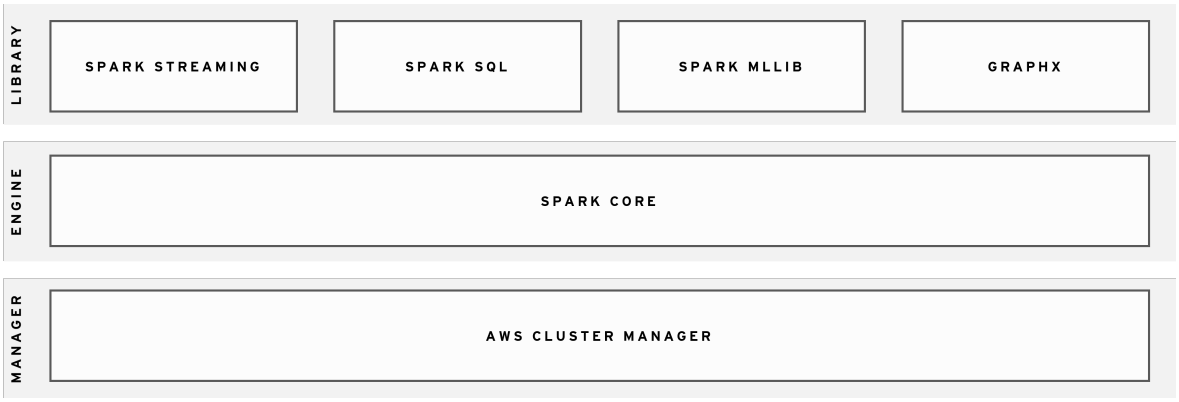

Figura 6.1: Spark Stack

- *Spark Core*: è il motore di esecuzione della piattaforma Spark, al di sopra del quale sono costruite tutte le altre funzionalità.

6.1.2 Formato parquet

Tutti i file risultanti dalle fasi di *pre-processing* sono stati salvati in formato *parquet* all'interno del DataLake.

Il *parquet* è un formato *open source* e con licenza Apache, che organizza i dati in colonne, a differenza del più comune *csv*, che invece li organizza per righe. Esso presenta numerosi vantaggi rispetto al suo concorrente, in quanto, a parità di dati, occupa molto meno spazio in memoria perché esegue una compressione adeguata per ogni colonna; ciò non è possibile per la memorizzazione per righe, che, normalmente, contiene più tipi di dati.

La minor occupazione di disco è un aspetto importantissimo perché si mappa in maniera diretta in una riduzione dei costi di archiviazione su S3 per il cliente, ma non è l'unico. A questo, infatti, si aggiunge l'estrema efficienza di questo particolare formato, dovuta all'architettura di archiviazione per colonne, che permette di saltare rapidamente i dati non rilevanti, velocizzando notevolmente le interrogazioni.

6.2 Pre-processing del Dataset

I dati grezzi, ovvero nello stato nel quale vengono inviati dai sensori al Data Lake, chiaramente non possono essere direttamente forniti in input a un algoritmo di apprendimento automatico. Proprio per questo motivo, quindi, la prima fase dello sviluppo dello Use Case è quella di elaborare il dataset (presentato al Capitolo 4), sottoponendo i dati ad una serie di manipolazioni, atte a prepararli alle successive fasi progettuali.

Oltre alle più tipiche operazioni di gestione dei valori nulli, di quelli fuori *range* e la rimozione di eventuali duplicati, sono state definite e applicate ai dati delle funzioni di elaborazione più complesse e specifiche per il dataset in questione.

Le prossime sottosezioni forniranno una breve spiegazione delle principali operazioni di *pre-processing* eseguite.

6.2.1 Ricongiungimento dei pacchetti

Da un punto di vista teorico, i sensori campionano i dati ogni 10 secondi, quindi con una frequenza di 10 Hz. Da un punto di vista pratico, però, i campioni non vengono acquisiti tutti insieme, ma divisi in due pacchetti. Tra la prima e la seconda *tranche* di campionamenti

intercorre circa 1 secondo; in sostanza, ciò comporta che vengano scritti sul Data Lake due record per ogni evento di acquisizione.

Per poter utilizzare questo dataset, è stato, quindi, necessario riunire i due pacchetti di campioni in modo da ottenere un solo record per ogni evento, facendo attenzione a non unificare record successivi facenti riferimento a sessioni di guida diverse.

Come prima cosa, pertanto, è stata definita una logica per l'identificazione dei *trip*, ovvero le sessioni di guida. Utilizzando la libreria PySpark di Python sono stati raggruppati i record per vin e ordinati per istante di acquisizione crescente; un'iterazione sulle righe ha, poi, consentito di calcolare la differenza tra *time* di righe successive e di assegnare i record a sessioni di guida diverse, qualora tale differenza risultasse superiore ad una soglia predefinita.

6.2.2 Gestione anomalie invio

Per gestire anche delle anomalie del dataset, riguardanti l'invio di acquisizioni aggiuntive in istanti non previsti, la differenza di *time* viene anche utilizzata per unificare righe successive, dello stesso *trip*, acquisite a meno di 8 secondi una dall'altra. L'unica accortezza presa in questo caso è stata quella di verificare che i campi "valorizzati" in entrambe le righe assumessero lo stesso valore. Solo in questo caso, infatti, è stata eseguita la fusione dei record; in caso contrario, questi sono stati mantenuti separati, continuando ad essere considerati come due eventi di acquisizione distinti.

6.2.3 Identificazione e calcolo delle feature

Da un'analisi approfondita dei dati a disposizione e da uno studio condotto in rete, si è arrivati a definire quali potessero essere le principali variabili da monitorare per determinare la necessità di eseguire operazioni di manutenzione sui veicoli. Nello specifico, si è deciso di attenersi alla valutazione di informazioni in grado di evidenziare un uso spinto del veicolo, al quale sarà inevitabilmente associata una maggiore usura dei componenti e, di conseguenza, la necessità di eseguire più di frequente interventi di manutenzione.

Sulla base di tali ragionamenti si è, quindi, arrivati a identificare le variabili di interesse per lo sviluppo delle *feature* da fornire in input agli algoritmi di apprendimento automatico.

In letteratura, una delle informazioni più rilevanti per eseguire manutenzione predittiva sull'intero veicolo è lo stile di guida dell'utente. Sebbene possa sembrare secondario, questo aspetto è, invece, di fondamentale importanza, in quanto l'usura dell'auto e dei suoi componenti dipende in maniera predominante da quanto e come questa venga utilizzata. Non servono, infatti, competenze meccaniche per intuire che, anche a parità di km percorsi e di contesto di utilizzo, lo stato di salute di un veicolo, il cui conducente ha l'abitudine di scattare al semaforo e percorrere le strade al massimo della velocità consentita, per poi frenare bruscamente, non potrà mai essere lo stesso di uno che, invece, viene guidato unicamente a velocità moderate.

La determinazione dello stile di guida, però, è un processo tutt'altro che banale e, soprattutto, si avvale di alcune informazioni ignote riguardanti il conducente, il consumo istantaneo e la tipologia di strada percorsa (città, campagna, montagna). Per cercare in qualche modo di tener conto di questo aspetto dello stile di guida, basandosi però sulle informazioni a disposizione, si è pensato di costruire delle *feature* riguardanti le accelerazioni, le frenate e le curve brusche, l'eventuale attivazione del *launch control*, che consente di effettuare uno sprint in partenza, o del controllo dinamico della stabilità. Tutto questo tenendo conto anche dei km del viaggio e della quantità di carburante consumato complessivamente.

Come dichiarato nella Sezione 1.4, riguardante gli obiettivi aziendali, il cliente pone molta attenzione nei confronti dell'ambiente, motivo per il quale il modello di auto in analisi, uscito

da circa un anno e mezzo, è un veicolo di tipo ibrido. Questo, purtroppo, ha complicato il tentativo di definire delle *feature* che coinvolgessero informazioni utili a non perdere di vista l'aspetto dei consumi di carburante. La scelta fatta, in questo caso, è stata quella di integrare anche le informazioni relative alla batteria del veicolo.

Come evidenziato nella Tabella 4.1, però, si hanno a disposizione due campi i cui nomi sono riferibili al concetto di batteria, ovvero *battery level* e *battery level p*. Sebbene dalle descrizioni delle variabili a nostra disposizione i due campi dovrebbero rappresentare lo stesso concetto, con l'unica differenza che *battery level p* dovrebbe esprimerlo in percentuale, in realtà, i valori non corrispondono. Pertanto, assumendo che questi campi potrebbero far riferimento alle due batterie del veicolo, quella posseduta da tutte le auto a combustione, e quella propria dei veicoli elettrici, entrambi sono stati utilizzati per la definizione delle *feature*.

Chiaramente, si è deciso di non tralasciare l'usura del motore, cuore pulsante dell'intero sistema meccanico. In particolare, per tenere conto dello sforzo prodotto, sono stati considerati il numero di giri al minuto e la potenza istantanea erogata, insieme alla temperatura del liquido di raffreddamento e a quella dell'olio. Il monitoraggio di questi ultimi due parametri è particolarmente importante e, allo stesso tempo, agevolato dal fatto che esistono dei *range* predefiniti proprio a livello meccanico; resta, però, da considerare che si stanno trattando delle auto sportive e che, quindi, le classiche soglie di allarme potrebbero variare rispetto allo standard.

Per quanto riguarda le ruote, si è deciso tenere conto soltanto della temperatura del giunto a quattro ruote motrici, al fine di monitorarlo per rilevare eventuali surriscaldamenti anomali.

La Tabella 6.1 elenca le *feature* che si è ritenuto opportuno definire, spiegando anche come sono state ottenute, tenendo sempre conto che sono state calcolate a livello di *trip*.

Features	**Descrizione modalità di calcolo**
Accelerazioni longitudinali e laterali brusche	Conteggio dei valori positivi di *acc longitudinal* e *acc lateral* al di sopra di una soglia predefinita
Decelerazioni longitudinali e laterali brusche	Conteggio dei valori negativi di *acc longitudinal* e *acc lateral* al di sotto di una soglia predefinita
Accelerazione media e massima per modalità di guida	Per ogni modalità di guida indicata da *driving mode*, quindi Corsa, Strada e Sport, sono state calcolate la media e il massimo dei valori positivi di *acc longitudinal* e *acc lateral*
Decelerazione media e massima per modalità di guida	Per ogni modalità di guida indicata da *driving mode*, quindi Corsa, Strada e Sport, sono state calcolate la media e il massimo dei valori negativi di *acc longitudinal* e *acc lateral*
Velocità media	Calcolo della media di *speed*
Massimo, minimo e media del livello di carburante nel serbatoio	Calcolo del valore medio, minimo e massimo di *fuel level* e *fuel tank*
Km percorsi	Differenza tra il valore di *km tachimeter* registrato alla fine e all'inizio del *trip*
Livello minimo e medio della batteria	Calcolo del valore medio e massimo di *battery level* e *battery level p*
Potenza minima, media e massima del motore	Calcolo del valore minimo, medio e massimo di *power instant*

Numero minimo, medio e massimo di giri del motore al minuto	Calcolo del valore minimo, medio e massimo di *rpm*
Attivazione del controllo dinamico della stabilità	Posto ad 1 se, almeno una volta all'interno del *trip*, si è attivato l'*esp*, che consente di riprendere il controllo della vettura a seguito di una sbandata
Minimo, massimo e media del valore della pressione esercitata sul pedale dell'acceleratore	Calcolo del valore minimo, medio e massimo di *throttle*
Pressione elevata del pedale dell'acceleratore	Conteggio di quante volte nel *trip* la variabile *throttle* ha assunto un valore compreso tra quello medio e massimo (definiti in precedenza).
Pressione minima, media e massima dei freni	Calcolo del valore minimo, medio e massimo di *brake pressure*
Pressione freni negativa	Conteggio di quante volte nel *trip* il *brake pressure* ha assunto un valore negativo
Sterzate brusche	Conteggio di quante volte il valore di *angular speed* è stato superiore al terzo quartile calcolato su tutti i dati del mese
Coppia di sterzo media e massima trasmessa dal conducente al volante	Calcolo del valore medio e massimo di *torque*
Coppia di sterzo elevata	Conteggio di quante volte *torque* ha assunto un valore compreso tra quello medio e massimo (definiti in precedenza)
Temperatura media e massima del liquido di raffreddamento del motore	Calcolo del valore massimo e medio di *coolant temperature*
Temperatura corretta del liquido di raffreddamento del motore	Conteggio di quante volte il valore di *coolant temperature* ha assunto un valore compreso tra 80 e 90
Temperatura elevata del liquido di raffreddamento del motore	Conteggio di quante volte il valore di *coolant temperature* ha assunto un valore superiore a 90
Tempo per il quale è stata registrata una temperatura corretta del liquido di raffreddamento del motore	Somma di quanti secondi per ogni *trip* si è registrato un valore corretto (secondo la definizione fornita in precedenza) di temperatura del liquido di raffreddamento del motore
Tempo per il quale è stata registrata una temperatura elevata del liquido di raffreddamento del motore	Somma di quanti secondi per ogni *trip* la temperatura del liquido di raffreddamento del motore è stata elevata (secondo la definizione fornita in precedenza)
Temperatura media e massima dell'olio del motore	Calcolo del valore medio e massimo di *oil temperature*
Temperatura elevata dell'olio del motore	Conteggio di quante volte il valore di *oil temperature* ha assunto un valore compreso tra quello medio e massimo (definiti in precedenza)

Tempo per il quale è stata registrata una temperatura elevata dell'olio del motore	Somma di quanti secondi per ogni *trip* la temperatura dell'olio del motore è stata elevata (secondo la definizione fornita in precedenza)
Temperatura media e massima dell'olio nel giunto a 4 ruote motrici	Calcolo del valore medio e massimo di *4 wheel drive oil temp coupling*
Temperatura elevata dell'olio nel giunto a 4 ruote motrici	Conteggio di quante volte il valore di *4 wheel drive oil temp coupling* ha assunto un valore compreso tra quello medio e massimo (definiti alla riga precedente)
Tempo per il quale è stata registrata una temperatura elevata dell'olio nel giunto a 4 ruote motrici	Somma del numero di secondi per ogni *trip* in cui la temperatura dell'olio nel giunto a 4 ruote motrici è stata elevata (secondo la definizione riportata alla riga precedente)

Tabella 6.1: Descrizione delle *feature* calcolate

Considerando che la richiesta del cliente è quella di poter effettuare manutenzione predittiva di mese in mese, l'input della pipeline di *Machine Learning* sarà, di volta in volta, soltanto una porzione del dataset relativa ad uno specifico mese. Quest'ultimo avrà un *record* per ogni sessione di guida registrata in quell'arco temporale, contenente i valori di tutte le *feature* calcolate.

6.3 Pipeline di apprendimento automatico

Come discusso nel precedente capitolo, dopo aver valutato vari approcci di *Machine Learning*, tenendo conto dei dati e delle conoscenze di dominio a disposizione, si è deciso di adottare una tipologia di approccio mista. Questa combina un algoritmo di apprendimento non supervisionato, in particolare di *clustering*, con uno di apprendimento supervisionato, in particolare un classificatore binario, che consentirà, in definitiva, di prevedere quali e quanti veicoli avranno bisogno di manutenzione nel mese prossimo.

6.3.1 Standardizzazione delle feature

Il dataset, al suo interno, presenta delle *feature* i cui *range* di valori sono molto diversi l'uno dall'altro, anche perché facenti riferimento ad unità di misura molto differenti tra di loro. Queste differenze causano problemi a molti modelli di apprendimento automatico, soprattutto quelli che si basano sul calcolo della distanza, come quelli di *clustering*. In particolare, il problema che si genera è determinato dal fatto che le *feature* con *range* elevati avranno una maggiore influenza sul *clustering*, polarizzando, quindi, l'esito dell'algoritmo.

Al fine di garantire, in maniera più corretta, che le *features* contribuiscano in egual modo alla definizione del modello, è stato necessario applicare la tecnica della standardizzazione.

Standardizzare un vettore di *feature* significa, innanzitutto, calcolarne la media e deviazione standard, per sottrarre la prima da ogni elemento e dividere il risultato per la seconda. In questo modo, ogni vettore di *feature* avrà media pari a 0 e deviazione standard pari a 1; ciò garantirà di evitare l'introduzione di *bias* a causa delle sopracitate differenze di *range*.

6.3.2 Apprendimento non supervisionato

Come già anticipato, il primo algoritmo che compone la pipeline di *Predictive Maintenance* è di tipo non supervisionato; in particolare, si parla di *clustering*, mirato alla determinazione della variabile target per il successivo algoritmo supervisionato.

Clusterizzare un dataset significa segmentarlo, suddividendo i dati che lo compongono in sottoinsiemi, chiamati, appunto, cluster. Un cluster è un insieme di oggetti che presentano tra loro delle similarità, ma che, per contro, presentano dissimilarità con gli oggetti di altri cluster. Essendo una tecnica di apprendimento non supervisionato, questo processo non prevede una fase di apprendimento durante la quale vengono forniti esempi da cui imparare. Sono proprio le caratteristiche intrinseche dei dati a guidare la loro suddivisione nei diversi sottoinsiemi, eseguita tramite l'applicazione di un insieme di tecniche statistiche multivariate.

Per lo sviluppo dello Use Case di *Predictive Maintenance* si è scelto di utilizzare l'algoritmo K-means, uno degli algoritmi di *clustering* più diffuso e performante.

6.3.2.1 K-means

Nonostante la semplicità della sua logica, K-means è uno degli algoritmi di *clustering* più conosciuti e utilizzati, grazie agli ottimi risultati a cui è in grado di arrivare. Il suo funzionamento è iterativo e mira a partizionare i dati in sottoinsiemi, detti *cluster*, la cui definizione passa per la progressiva ottimizzazione della posizione dei centroidi di ogni *cluster*.

Il centroide è un punto che media, da cui il termine *"means"* che compone il nome dell'algoritmo, le distanze tra tutti i dati assegnati al *cluster* che esso rappresenta. Costituisce, quindi, una sorta di baricentro del *cluster* ed in generale, proprio per le sue caratteristiche, non è uno dei punti del dataset.

6.3.2.1.1 Metodo ottimizzativo dei parametri

Tipicamente, gli algoritmi non supervisionati fanno inferenze da una serie di dati, utilizzando solo i vettori di input senza fare riferimento ad alcuna informazione aggiuntiva. Il K-means, però, chiede in input non soltanto il dataset, ma anche una serie di parametri, dai quali dipenderà fortemente la bontà del risultato finale. La prima cosa da fare, pertanto, è quella di determinare il valore ottimale dei parametri da fornire in input all'algoritmo, ovvero definire il valore di K (da cui il nome del metodo), che rappresenta il numero di cluster nei quali si vuole suddividere il dataset, e quello del *seed*, utilizzato per generare un numero *randomico* che influenza l'inizializzazione delle posizioni dei centroidi.

Il risultato dell'algoritmo K-means è fortemente dipendente dalla scelta di tali parametri. Al fine di garantire il migliore risultato possibile, è stato sviluppato un metodo ottimizzatore dei parametri, utile a determinare la migliore combinazione possibile dei valori di *K* e *seed*, valutata in termini di *silouhette*. Quest'ultima è l'indicatore di performance utilizzato per la *Cluster Analysis*; essa misura la coesione, e quindi la similarità, degli oggetti di uno stesso cluster, nonché la separazione, e quindi la dissimilarità, rispetto a quelli di altri cluster. La formula che consente di determinarla è una diretta traduzione matematica della definizione fornita ed è pari al rapporto tra la differenza della distanza media extracluster e quella intracluster, e il massimo tra le due distanze. La sua modalità di calcolo comporta che i valori assumibili varino tra -1 e 1; in particolare, tanto maggiore sarà il suo valore tanto migliore sarà il risultato ottenuto.

Il metodo ottimizzativo progettato esegue ripetutamente l'algoritmo di *clustering*, facendo variare ogni volta il valore *K* all'interno di un *range* predefinito; per ogni valore di *K*, viene quindi variato anche quello del *seed*. Al termine dell'esecuzione, per ogni *K*, vengono forniti i

valori di *silouhette* ottenuti al variare di *seed* e, solo per il migliore di questi, vengono anche stampate la numerosità di ogni cluster e la posizione dei corrispettivi centroidi.

Dall'analisi degli output è, quindi, possibile determinare i valori di K e di *seed* che garantiscano il migliore risultato, in modo da poterli, poi, rendere parametri statici, da utilizzare per l'applicazione del K-means su tutti i dataset mensili.

Nella Sezione 7.1.1 sono riportati i risultati del metodo.

6.3.2.1.2 Funzionamento dell'algoritmo

Definiti correttamente i parametri, è stato possibile applicare l'algoritmo K-means che, dal punto di vista operativo, si compone dei seguenti step:

1. si scelgono in modo casuale, sfruttando il parametro seed, le posizioni iniziali per i K centroidi, non coincidenti;
2. si calcola la distanza di ogni punto del dataset rispetto ad ogni centroide;
3. ogni punto del dataset viene assegnato al cluster rappresentato dal centroide più vicino;
4. si ricalcola la posizione di ogni centroide come media, per ogni cluster, delle posizioni di tutte le *feature* ad esso assegnate, al fine di affinarla progressivamente;
5. si itera, ripetendo i punti 3 e 4, fino a quando la posizione del centroide non si stabilizza, non subendo più ulteriori modifiche.

Nella Sezione 7.1.2 è riportato il risultato dell'esecuzione della *Cluster Analysis* con l'algoritmo di K-means, sulla base del quale è stato possibile definire e applicare la logica di determinazione della variabile target dell'algoritmo supervisionato.

6.3.3 Algoritmo supervisionato

Per arrivare a definire la necessità di fare manutenzione su un determinato vin, si è deciso di definire un modello di classificazione binaria, che produrrà, quindi, uno 0 o un 1 in risposta ad ogni dato di input ricevuto. Ogni modello di apprendimento automatico supervisionato necessita, tuttavia, di una fase di training, che consenta ad esso di apprendere la corretta logica di etichettatura dei dati a partire da un set di dati già etichettato.

6.3.3.1 Definizione della logica di etichettatura del dataset di training

Per definire l'etichetta da assegnare ad ogni dato si è deciso di partire dai vettori che descrivono le posizioni dei centroidi nello spazio delle *feature* e dalla loro posizione media. I centroidi, infatti, fungono da rappresentanti dell'intero cluster ai quali sono associati; pertanto, si è ritenuta una idea valida quella di sfruttare la conoscenza delle loro posizioni per la definizione della logica di etichettatura binaria dei dati, dove 1 indica la necessità di manutenzione e 0 la mancanza di tale necessità.

Nello specifico, per ogni centroide, sono state confrontate le sue componenti, rispetto alla varie *feature*, con quelle del vettore della posizione media. Dalla valutazione di ogni componente in relazione al significato della *feature* associata, si è arrivati all'assegnazione di uno 0 o di un 1 ad ogni *feature* di ogni *trip*. Ad esempio, considerando *acc longitudinal brusche*, che indica il numero di accelerazioni al di sopra di una determinata soglia rilevate per ogni *trip*, si è assegnato valore 1 al centroide la cui componente è risultata superiore alla media, 0 in caso contrario. Chiaramente, ogni *feature* è stata valutata singolarmente; alcune, infatti, per

via di ciò che rappresentano, necessitano di una logica contraria, quindi 0 se superiori alla media e 1 altrimenti.

Dall'applicazione di questa logica al DataFrame dei centroidi si è arrivati ad ottenere un vettore per ogni cluster avente un numero di righe pari al numero del *features* e i cui valori possono essere pari unicamente a 0 o 1. A questo punto sono state raggruppate le *feature* in 3 macroaree (esempio riportato in Tabella 6.2), ovvero Alert, Consumi e Guida, con l'intento di sfruttare i vettori definiti per arrivare a ridurne il numero di righe a 3, ovvero una per macrogruppo.

Macroaree	**Features**	**Cluster_1**	**Cluster_2**	**Cluster_3**
M_1	feat_1	0	1	1
	feat_3	1	0	1
M_2	feat_2	0	0	1
	feat_4	1	1	0
	feat_5	1	0	1
	feat_6	1	1	1

Tabella 6.2: Tabella esemplificativa della prima parte del processo di etichettatura dei centroidi

A determinare il valore 0 o 1 per ogni cluster sono i vettori definiti in precedenza; in particolare, il valore sarà pari a 1 se almeno l'80% delle *feature* del macrogruppo di quel centroide assume tale valore, mentre sarà pari 0 altrimenti (esempio in Tabella 6.3).

Da qui ad ottenere l'etichetta definitiva per il cluster il passaggio è stato breve; infatti, è stato sufficiente assegnare a quest'ultimo l'etichetta più frequente (esempio in Tabella 6.3).

Macroaree	**Cluster_1**	**Cluster_2**	**Cluster_3**
M_1	0	0	1
M_2	1	0	1
Label finale	0	0	1

Tabella 6.3: Tabella esemplificativa delle ultime due fasi del processo di etichettatura dei centroidi, con riferimento ai valori in Tabella 6.2

Con una serie di passaggi piuttosto intricati, si è, quindi, arrivati ad assegnare 0 o 1 ad ogni cluster; tale etichetta è stata, poi, applicata anche a tutti i dati, ovvero i *trip*, contenuti nei cluster. L'interesse, però, è quello di arrivare a prevedere la necessità di manutenzione sui veicoli; pertanto, l'etichettatura a livello di *trip* non è in realtà ancora sufficiente. Per evitare, però, di aggregare ulteriormente le *feature*, introducendo inevitabilmente ulteriore perdita di dettaglio, si è deciso di classificare a livello di *trip* e di definire l'etichetta di vin, solo a valle del processo di classificazione. Anche in questo caso, essa sarà pari alla label più frequente nelle sessioni di guida di quel vin, registrate nell'arco temporale di un mese.

La logica di *labeling* progettata consente, quindi, di indicare il veicolo come bisognoso di manutenzione solo se effettivamente tale necessità risulta dall'analisi complessiva di tutti i *trip* mensili; in questo modo, quindi, non sarà un'unica guida più sfrenata o più moderata del normale utilizzo a deviare la corretta etichettatura del vin.

Dal punto di vista dell'algoritmo di classificazione, si è scelto di utilizzare XGBoost per via dell'elevata robustezza e delle ottime prestazioni che è in grado di garantire.

6.3.3.2 Extreme Gradient Boosting - XGBoost

Extreme Gradient Boosting, o XGBoost, è un algoritmo di *ensemble* di *Machine Learning* basato sugli alberi decisionali. Rispetto agli altri algoritmi supervisionati consente di ottenere migliori risultati utilizzando meno risorse computazionali e impiegando meno tempo di calcolo, grazie ad una serie di ottimizzazioni, tra le quali, la parallelizzazione dei processi.

Sviluppato come un progetto di ricerca all'Università di Washington e presentato in una conferenza nel 2016, la sua validità è stata immediatamente riconosciuta, tanto che poco dopo il suo sviluppo e il rilascio iniziale, XGBoost è diventato un metodo di riferimento per i problemi di classificazione e regressione. Esso si comporta bene nelle competizioni di apprendimento automatico grazie a un robusta gestione della maggior parte dei tipi di dati, relazioni, distribuzioni, e alla varietà di iperparametri che è possibile regolare con precisione. Tali caratteristiche hanno consentito ad XGBoost di essere spesso utilizzato come componente chiave dalle soluzioni vincenti nelle competizioni di *Machine Learning*, superando persino l'uso delle reti neurali.

Il fattore più importante dietro il successo di XGBoost è la sua scalabilità su ambienti distribuiti, dovuta a diversi importanti sistemi e ottimizzazioni algoritmiche, che consentono ad esso di superare, di più di dieci volte, la velocità di altre soluzioni esistenti, progettate per lavorare su singola macchina.

6.3.3.2.1 Funzionamento dell'algoritmo

Il funzionamento dell'algoritmo è piuttosto complesso, per cui, in questo caso, non si scenderà nei dettagli, ma verranno fornite solo informazioni più generali.

XGBoost un *ensamble model* basato su alberi decisionali; gli alberi vengono aggiunti uno alla volta all'insieme e adattati per correggere gli errori di previsione effettuati dai modelli precedenti. Ogni albero, infatti, riceve in input l'output del precedente, al quale sono però stati variati i pesi, aumentando quello degli oggetti classificati erroneamente e diminuendo quello degli altri. In questo modo, il nuovo albero si concentrerà maggiormente nella corretta classificazione degli oggetti per i quali era stata sbagliata la previsione in precedenza.

La previsione finale fornita da XGBoost per ogni oggetto è determinata dalla previsione più frequente, nel caso della classificazione, e dalla media delle previsioni, nel caso della regressione.

Quando si utilizzano algoritmi di apprendimento automatico che operano delle scelte stocastiche, come nel caso degli alberi decisionali, è buona pratica valutarli calcolando la media delle loro prestazioni su più esecuzioni.

6.3.3.3 K-Fold Cross Validation

Per valutare la bontà del modello di classificazione è stata utilizzata la K-Fold Cross Validation, una tecnica statistica per la validazione del modello che fornisce risultati più affidabili rispetto alla classica suddivisione dei dati nei tre insiemi di *train*, *test* e *validation*.

Questa procedura di validazione consiste nel suddividere il dataset etichettato in un numero K di parti (in letteratura i valori più comuni sono 5 e 10), utilizzarne ciclicamente una diversa come dataset di *validation* e le restanti per il *training*. Questo significa eseguire per K volte il processo di *training* del modello, per poi validarlo, di volta in volta, su un set di dati diverso.

Le performance complessive saranno determinate dalla media della performance di ogni modello; esse saranno espresse in termini di *Accuracy*, *Precision* e *Recall*.

6.4 Mantenimento della qualità dei processi e dell'efficacia degli algoritmi

Come già detto, tutto lo sviluppo dello Use Case è stato pensato per essere eseguito mensilmente; questo significa che, al primo di ogni mese, il cliente dovrà ottenere le previsioni relative a quali e quanti vin potrebbero richiedere interventi nei successivi 30 giorni, formulate sulla base dei dati dei mesi precedenti. Per poter garantire il corretto funzionamento e la massima efficacia della pipeline di *Machine Learning* nel tempo, è stata progettata un'apposita schedulazione dei processi.

Il job principale si compone di quattro job sequenziali, di cui due costituiscono il cuore della *Predictive Maintenance* e sono, chiaramente, quelli associati ai due algoritmi, quindi il job di *clustering* e quello di *classification*.

Ogni primo del mese si attiva il job di *pre-processing*, che legge da S3 tutti i dati acquisiti nell'arco del mese precedente e li sottopone a tutte le fasi di elaborazione descritte nella Sezione 6.2. Al suo termine si attiva il job di *clustering* che, al suo interno ne include uno di orchestrazione del modello, che, come prima cosa, verifica a quale mese sono associati i dati ricevuti in input. La ridefinizione del modello, infatti, viene eseguita solo due volte all'anno, a distanza di sei mesi una dall'altra; negli altri casi ci si limita soltanto ad andare in *scoring*, applicando l'ultimo modello disponibile ai dati del nuovo mese, allo scopo di monitorare la variazione dei cluster rispetto ai centroidi identificati in precedenza.

Ogni nuova disponibilità di dati processati, porta all'attivazione del job di etichettatura; il risultato sarà, poi,x salvato sempre su S3, al fine di poterlo utilizzare per future fasi di ri-addestramento del modello di classificazione.

Il job di classificazione, invece, va in *rolling* ogni mese; questo significa che viene riaddestrato il classificatore sulla base delle sei mensilità precedenti il nuovo mese acquisito. Il modello ottenuto viene, poi, applicato a quest'ultimo, in modo da produrre l'output desiderato, ovvero i vin che dovranno essere richiamati per la manutenzione.

L'immagine in Figura 6.2, riporta l'intero processo in forma schematica.

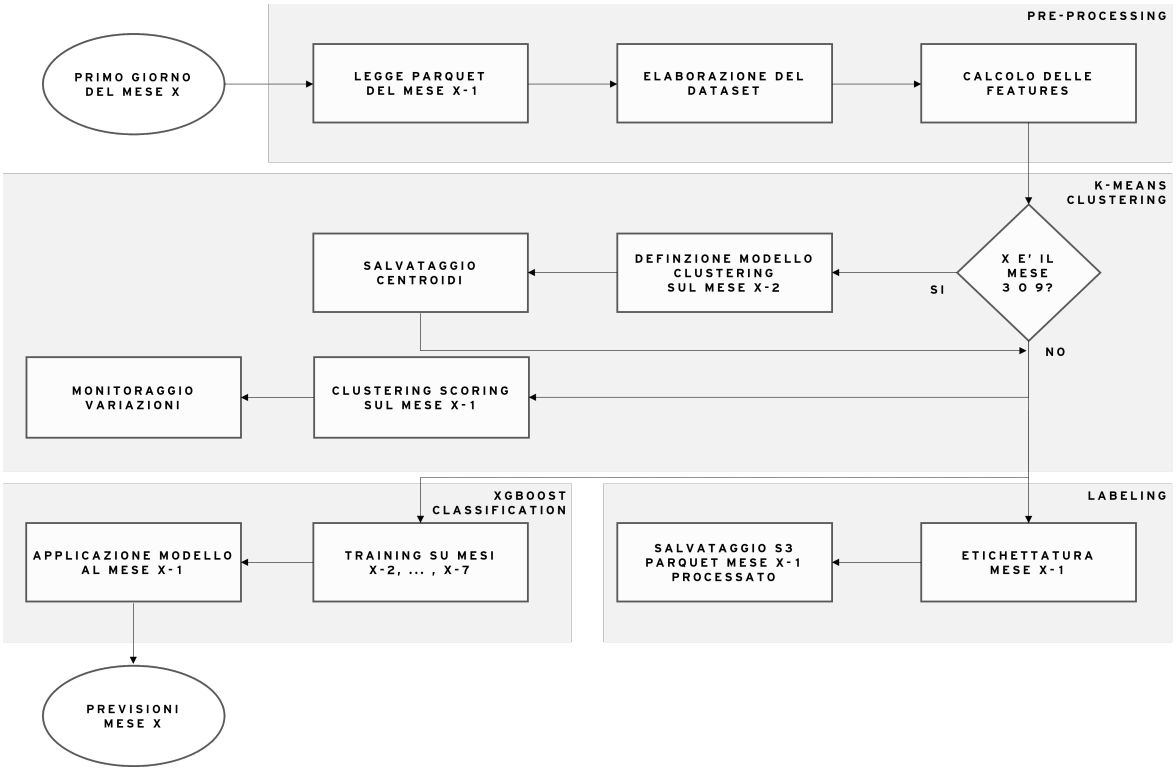

Figura 6.2: Schedulazione per il mantenimento della qualità dei processi e dell'efficacia degli algoritmi

CAPITOLO 7

Risultati sperimentali

Questo capitolo chiude la presentazione dello Use Case, fornendo tutti i risultati del caso. Si partirà dai valori di K e seed ottenuti tramite l'applicazione del metodo ottimizzativo dei parametri, per arrivare all'effettivo risultato di K-means. Successivamente, verranno forniti i valori delle metriche di valutazione del modello di classificazione prodotto da XGBoost e, in aggiunta, anche l'accuratezza delle previsioni finali, valutata tramite l'utilizzo di un dataset, fornito a valle dell'intero processo di sviluppo e contenente lo storico di tutti gli interventi di manutenzione eseguiti sui veicoli.

7.1 Algoritmo non supervisionato - Clustering con K-means

7.1.1 Metodo ottimizzativo dei parametri

Come spiegato nella Sezione 6.3.2.1, prima di arrivare all'applicazione dell'algoritmo K-means è stato necessario definire opportunamente i parametri *K* e *seed*, ed è stato sviluppato un apposito metodo per farlo, presentato nella Sezione 6.3.2.1.

Dall'analisi dei suoi output, è stato possibile porre a 4 il valore di *K*, indicante il numero di cluster in cui suddividere i dati, e a 250 quello di *seed*, utilizzato per generare un numero randomico che influenza l'inizializzazione delle posizioni dei centroidi.

Dovendo essere in grado di fornire previsioni mese per mese, la correttezza di parametri è stata testata eseguendo l'intero processo di ottimizzazione anche sui dataset di *features* calcolate su varie mensilità a disposizione. I risultati hanno continuato a confermare la correttezza di *K* e di *seed* indicati. Tuttavia, ci si riserva di continuare a monitorarli non appena si avranno a disposizione le acquisizioni dei mesi successivi a Febbraio 2022.

La Tabella 7.1 riporta il valore di silouhette ottenuto in relazione ai parametri indicati.

K	Seed	Silouhette
4	250	0.48

Tabella 7.1: Valori di *K* e *seed* prodotti dal metodo ottimizzativo, in relazione con la *Silouhette*

La Tabella 7.2 riporta la suddivisione dei dati di un mese per cluster, come si può osservare la distribuzione è buona in quanto piuttosto omogenea. Altre combinazioni di *K* e *seed*, invece mostravano un grosso sbilanciamento, con cluster contenenti poche decine di dati, o anche meno.

Cluster	Numerosità
0	2312
1	121
2	947
3	797

Tabella 7.2: Distribuzione dei trip mensili per cluster

7.1.2 K-means

Per poter visualizzare in maniera grafica il risultato dell'algoritmo di *clustering* è stato necessario ridurre la dimensionalità del dateset attraverso l'applicazione della *Pricipal Component Analysis* (PCA). Questa tecnica consente di "mappare" le *feature* in input in un numero ridotto di variabili artificiali, 2 in questo caso, cercando di mantenere invariato il contenuto informativo e preservando la stessa variabilità del dataset ricevuto in input.

L'immagine in Figura 7.1 riporta la rappresentazione dei custer ottenuti.

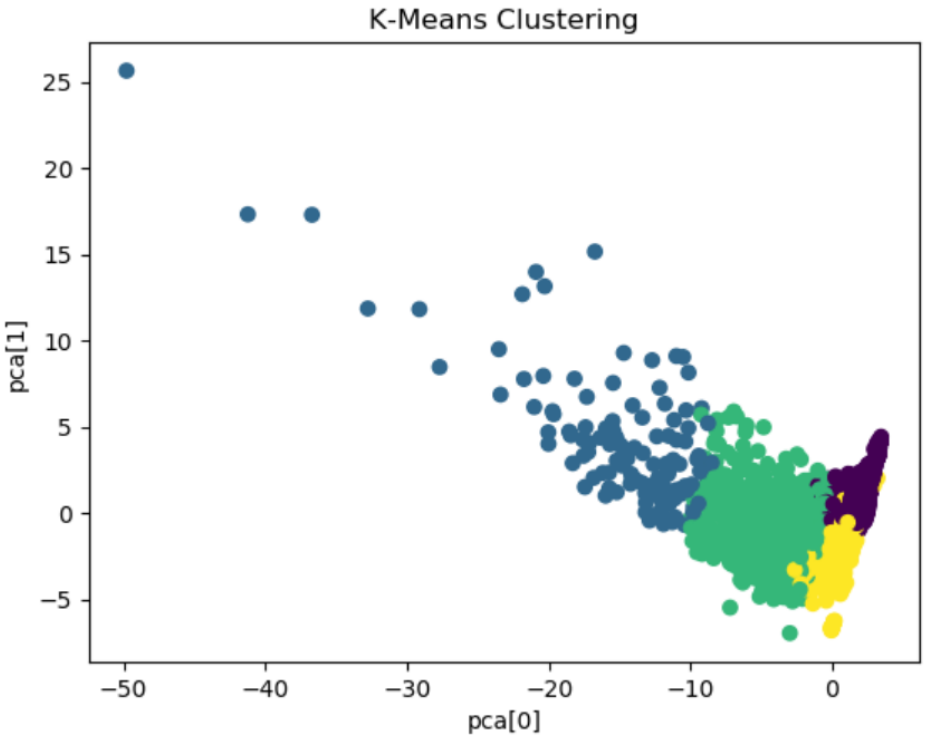

Figura 7.1: Cluster risultanti dall'applicazione di K-means

Come si può osservare, sono presenti alcuni *outlier*; pertanto, il DataFrame PySpark è stato dato in input ad un apposito metodo di rimozione degli *outlier*. Il risultato ottenuto da K-means eseguito a valle di questa ulteriore elaborazione, è riportato in Figura 7.2.

La Tabella 7.3 mostra, invece, le posizioni dei centroidi di ogni cluster, ottenute dopo un *rescaling*, effettuato moltiplicando le componenti relative ad ogni *feature*, con le rispettive deviazioni standard, per poi sommarle alla media, riportando, quindi, i valori alla situazione precedente la standardizzazione.

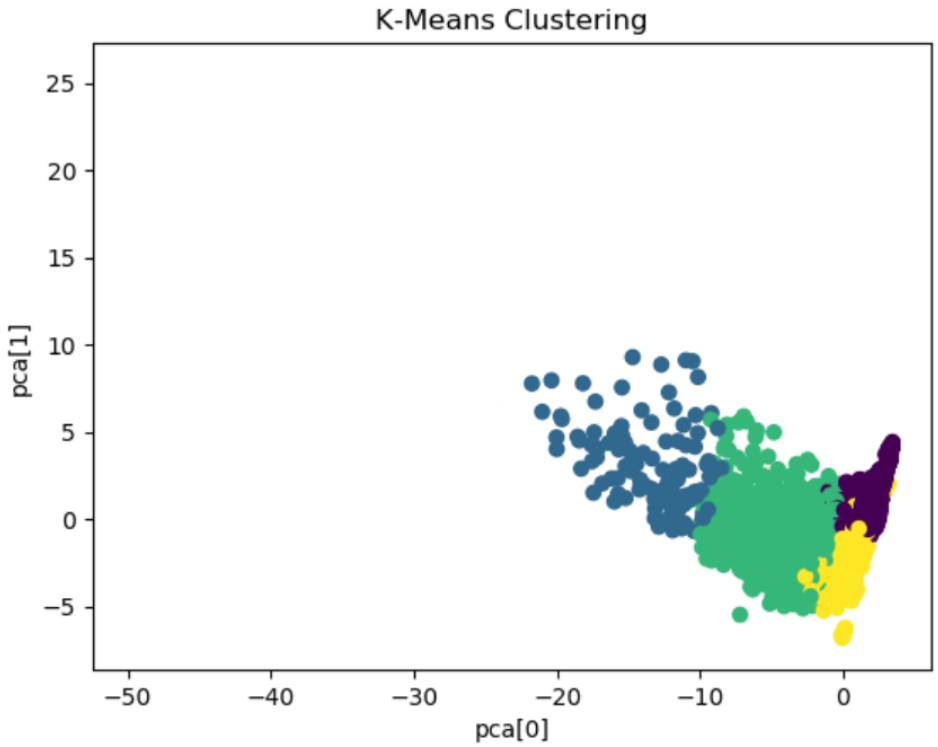

Figura 7.2: Cluster risultanti dall'applicazione di K-means a valle della rimozione degli outlier

Features	Centroide Cluster 0	Centroide Cluster 1	Centroide Cluster 2	Centroide Cluster 3
4 wheel drive oil temp coupling avg	35.219	35.219	35.219	35.219
4 wheel drive oil temp coupling avg-max	0.137	287.842	25.687	0.621
4 wheel drive oil temp coupling max	92.0	92.0	92.0	92.0
4 wheel drive oil temp coupling time between avg-max	0.763	887.074	97.475	0.751
acc longitudinal brusche	8.650E-4	0.231	0.003	1.318E-16
activ launch control sum	2.850	255.809	55.539	8.081
angular speed max-of-3q-month	1.437	197.735	47.066	5.954
battery level avg	11.990	12.699	11.934	12.284
battery level min	11.777	12.248	11.445	11.780
battery level p avg	66.472	80.067	74.733	75.001
battery level p min	66.176	75.537	73.347	74.516
brake pressure avg	0.934	1.278	2.015	4.685
brake pressure max	3.546	29.074	21.486	16.785
brake pressure min	0.096	-0.454	-0.412	0.885

brake pressure neg	1.018	164.512	28.254	2.809
coolant temperature 80-90	0.498	498.495	93.880	5.244
coolant temperature >90	0.065	72.900	12.442	0.289
coolant temperature avg	84.658	84.658	84.658	84.658
coolant temperature avg-max	0.765	594.495	114.037 6.900	
coolant temperature max	117.0	117.0	117.0	117.0
coolant temperature time between 80-90	6.365	4545.818	913.046	61.804
coolant temperature time between >90	0.930	763.057	150.521	3.538
dec longitudinal brusche	-2.406E-17	0.024	0.005	-3.642E-17
esp status max	0.981	0.991	0.943	0.948
fuel level avg	5.977	56.053	48.487	66.976
fuel level max	6.097	65.793	52.596	67.691
fuel level min	5.795	42.710	44.292	65.716
fuel tank avg	38.298	64.404	52.388	76.779
fuel tank max	39.115	76.305	56.198	77.897
fuel tank min	37.707	50.884	48.189	75.849
km	-450.531	89.462	10.948	-2618.521
oil temperature 90-100	0.254	508.876	83.881	2.383
oil temperature >100	0.039	63.314	9.931	0.150
oil temperature avg	34.633	92.361	81.521	40.903
oil temperature avg-max	4.176	465.545	98.229	13.870
oil temperature max	119.0	119.0	119.0	119.0
oil temperature rms	73.700	2356.807	970.443	206.825
oil temperature std	1.143	9.528	14.241	4.699
oil temperature time between 90-100	2.865	5058.512	899.357	32.446
oil temperature time between >100	0.455	645.892	106.878	1.648
power instant avg	96.168	35.074	30.658	155.289
power instant max	117.797	483.704	235.094	219.306
power instant min	46.789	-3.552E-14	2.768	97.239
rpm avg	229.934	2095.245	1823.618	867.647
rpm max	334.324	5982.789	4516.820	1441.250
rpm min	137.584	155.433	461.934	515.726
speed avg	1.647	59.006	38.932	4.982
throttle avg	0.499	14.482	11.544	1.671
throttle avg-max	3.451	293.669	55.216	11.032

throttle max	2.557	84.148	62.051	11.022
throttle min	0.038	-1.526E-16	0.1895	0.021
torque avg	1.856	15.349	16.116	5.619
torque avg-max	1.829	275.900	54.270	7.666
torque max	10.792	283.750	217.808	53.671

Tabella 7.3: Tabella dei centroidi dei cluster

7.2 Apprendimento supervisionato - Classificazione con XGBoost

7.2.1 Train del modello

La fase di training del modello, come anticipato nella Sezione 6.4, è stata eseguita su 6 mensilità diverse, etichettate tramite il processo descritto nella Sezione 6.3.3.1, eseguito, di mese in mese, da un apposito job.

Utilizzando la *K-Fold Cross Validation* con *K*=10 e lo *shuffle* dei dati attivato, sono state ottenute le performance riportate in Tabella 7.4.

TRAIN		**VALIDATION**	
Metriche	**Score**	**Metriche**	**Score**
Accuracy	0.89	Accuracy	0.80
Precision	0.82	Precision	0.72
Recall	0.74	Recall	0.64

Tabella 7.4: Distribuzione dei trip mensili per cluster

I risultati descritti dalle metriche sono assolutamente positivi; si registra, infatti, un 80% di *accuracy*, che indica che il modello è in grado di formulare un elevato numero di previsioni corrette sul totale delle previsioni fornite. Ancora più rilevante è il 72% di *precision*, metrica che misura il numero di veri positivi sul totale dei classificati positivi; questo, infatti, significa che non vengono generati molti fasi positivi, e ciò è fondamentale per lo Use Case in questione, perché significherebbe richiamare al *dealer* un veicolo anche quando non c'è l'effettiva necessità. Ciò costituirebbe, chiaramente, un enorme problema a livello di business, perché non solo l'azienda si troverebbe a confrontarsi con un cliente scontento per la perdita di tempo, ma dovrebbe anche coprire le ore uomo investite dai meccanici, nella ricerca di un problema inesistente.

7.2.2 Utilizzo del modello per la previsione

Il modello definito su sei mensilità è stato, poi, applicato ai dati del mese subito successivo per prevedere quali vin avrebbero richiesto interventi di manutenzione nel mese seguente.

A valle della produzione dei primi risultati, il business ha consentito di confrontare le previsioni ottenute con lo storico del dataset di Claim e Warranty, contenenti, tra le altre cose, anche tutti gli interventi di manutenzione eseguiti sui veicoli prodotti dalla casa automobilistica. Questo confronto ha consentito di stimare la correttezza dell'intero processo sviluppato per lo Use Case e, in particolare, la correttezza delle *features* definite e della logica

di etichettatura, in quanto la correttezza dei modelli è già stata dimostrata dalle apposite metriche di valutazione.

Quello che si è evinto è che le previsioni hanno un'accuratezza media all'incirca pari al 72%. Tali risultati sono indubbiamente migliorabili ma già molto buoni.

CAPITOLO 8

Conclusioni e sviluppi futuri

Il modello di manutenzione predittiva descritto in questa tesi è soltanto un primo *deliverable* di un lungo e ambizioso progetto, tramite il quale ci si aspetta di arrivare a definire un modello di manutenzione predittiva definitivo, in grado di stimare, quanto più correttamente possibile, la necessità di richiamare il veicolo al *dealer*.

Come si è più volte evidenziato durante tutta la trattazione, la scarsa condivisione di informazioni da parte del business ha, inevitabilmente, polarizzato la scelta dell'approccio da seguire per lo sviluppo del modello. Questo, purtroppo, è un aspetto che, in fase iniziale di valutazione e prioritizzazione degli Use Case, non era stato messo in conto, supponendo che l'azienda, visto il grande interesse dimostrato e la continua richiesta di poter visionare i primi risultati, avrebbe fornito maggior supporto e avrebbe facilitato le riunioni del caso.

C'è, sicuramente, da evidenziare un aspetto, considerabile come una vera e propria *lesson learned*, ovvero che, per lo sviluppo dei prossimi Use Case di Big Data e *Advanced Analytics* con questo cliente, sarà necessario valutare, in fase di prioritizzazione, la disponibilità o meno di figure esperte di dominio in grado di fornire supporto in caso di necessità. Alla luce di quanto verificatosi in questo sviluppo, infatti, entrambe la parti hanno concordato la necessità di riunirsi, per stimare in maniera più dettagliata e puntuale la tipologia di conoscenza, esterna ai dati, che potrebbe essere richiesta per lo sviluppo dei successivi Use Case, al fine di identificare anticipatamente eventuali *gap* e individuare per tempo chi, all'interno dell'azienda cliente, possa essere in grado di sanarli opportunamente.

Certamente, come già evidenziato in fase di trattazione, questa assenza di informazione ha però consentito di applicare un approccio, indubbiamente più complesso, ma che ha comunque portato dei buoni risultati e consentito di estrarre, direttamente dai dati a disposizione, le informazioni utili alla definizione del modello.

Questo progetto sicuramente non è ancora arrivato ad un punto di arrivo, anzi è già possibile delinearne diversi sviluppi futuri; di seguito ne verranno proposti alcuni.

Come già detto, in fase di valutazione dei risultati, è stato consentito l'accesso al dataset di Claim e Warranty, che contiene tutti gli interventi di manutenzione, ordinaria e straordinaria, eseguiti sui veicoli della casa automobilistica. Per questioni legali che si stanno ancora trattando, il dataset non può ancora essere utilizzato per arricchire i dati a disposizione. Qualora, però, si arrivasse ad ottenere il consenso per l'integrazione di questi dati, questo dataset consentirebbe di conoscere con esattezza il tipo di guasto subito dai veicoli e quando questo si è verificato, nonché di stimare il *Mean Time Between Failure*, ovvero il tempo medio che intercorre tra due guasti, relativo non soltanto al modello, ma anche ai singoli vin.

La combinazione di queste informazioni non solo consentirà di migliorare l'attuale modello, ma permetterà anche di rivalutare l'adozione di un approccio puramente supervisionato. Dall'analisi delle variabili in relazione all'informazione del tipo di guasto e di quando questo si è verificato, ci si aspetta, infatti, di determinare una logica di etichettatura del dato, ma anche la possibilità di riconoscere le firme di guasto, che potrebbero aprire le porte del mondo della diagnosi dei guasti.

A livello tecnico, resta ancora un approccio di *Machine Learning* del tutto inutilizzato, ovvero quello Rule-Based. Sfruttando gli insight evidenziati dal modello, e continuando ad analizzare i dati di Connectivity, ci si augura di riuscire ad estrarre informazioni di valore sempre maggiore, che possano continuare ad arricchire la conoscenza sui dati, fino ad arrivare a definire delle regole precise per il popolamento della base di conoscenza.

Il dataset di Connectivity è stato fornito appositamente per lo sviluppo dello Use Case di *Predictive Maintenance*, pertanto su di esso non sono mai state condotte analisi di *Business Intelligence*. Il team Deloitte quindi, con approccio consulenziale, ha già proposto all'azienda cliente dei mock-up esemplificativi delle analisi descrittive che si potrebbero sviluppare tramite MicroStrategy. La casa automobilistica si è detta interessata ad alcune di queste proposte; ad esempio, lato Marketing e Commerciale, avere evidenza che alcune vetture viaggiano per lo più in modalità corsa, consentirebbe di proporre il pacchetto di giri in pista ad un prezzo più allettante. L'eventuale disponibilità all'utilizzo del dataset dei Claim e di quello di Warranty consentirebbe, in aggiunta, di analizzare i guasti, verificatisi all'interno della finestra temporale di garanzia dell'auto, in relazione ai costi. Questo permetterebbe di individuare la tipologia di guasti che si verificano più di frequente nel periodo di garanzia, e che quindi vanno ad intaccare pesantemente le finanze dell'azienda, di determinare i costi medi di garanzia di ogni veicolo, etc. Tutto questo, chiaramente, assumerebbe ancora più valore se incrociato con le previsioni di necessità di manutenzione prodotte dal modello sviluppato.

Pertanto, in parallelo al miglioramento del modello e allo sviluppo degli altri Use Case, verranno anche aggiunte queste ulteriori analisi di *Business Intelligence* che andranno ad integrare e arricchire quelle attuali.

Allo stato attuale, si è già attivato lo sviluppo di un secondo Use Case, sebbene questo non fosse il successivo in ordine di prioritizzazione, ovvero quello di GeoMarketing (Sottosezione 2.4.5.1), il cui scopo è quello di supportare il cliente nel determinare dove sia più opportuno aprire una nuova concessionaria. I dati di Connectivity, infatti, contengono anche la posizione GPS del veicolo; ciò ha consentito di anticipare lo sviluppo di questo nuovo Use Case, potendo anche sfruttare parte delle operazioni di preprocessing eseguite per lo Use Case di Predictive Maintenance, come, ad esempio, la definizione dei *trip*.

Bibliografia

ALI, Y. H. (2018), «Artificial Intelligence Application in Machine Condition Monitoring and Fault Diagnosis», in ACEVES-FERNANDEZ, M. A., curatore, «Artificial Intelligence», cap. 14, IntechOpen, Rijeka, URL https://doi.org/10.5772/intechopen.74932.

BYTTNER, S., RÖGNVALDSSON, T. e SVENSSON, M. (2011), «Consensus self-organized models for fault detection (COSMO)», *Engineering Applications of Artificial Intelligence*, vol. 24 (5), p. 833–839, URL https://www.sciencedirect.com/science/article/pii/S0952197611000467.

CARVALHO, T. P., SOARES, F. A. A. M. N., VITA, R., DA P. FRANCISCO, R., BASTO, J. P. e ALCALÁ, S. G. S. (2019), «A systematic literature review of machine learning methods applied to predictive maintenance», *Computers & Industrial Engineering*, vol. 137, p. 106 024, URL https://www.sciencedirect.com/science/article/pii/S0360835219304838.

CHEN, T. e GUESTRIN, C. (2016), «XGBoost: A Scalable Tree Boosting System», in «Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining», KDD '16, p. 785–794, Association for Computing Machinery, New York, NY, USA, URL https://doi.org/10.1145/2939672.2939785.

EHSANI, M., GAO, Y., LONGO, S. e EBRAHIMI, K. (2018), in CRC PRESS, curatore, «Modern Electric, Hybrid Electric, and Fuel Cell Vehicles», Third ed.

GIORDANO, D., GIOBERGIA, F., PASTOR, E., LA MACCHIA, A., CERQUITELLI, T., BARALIS, E., MELLIA, M. e TRICARICO, D. (2022), «Data-driven strategies for predictive maintenance: Lesson learned from an automotive use case», *Computers in Industry*, vol. 134, p. 103 554, URL https://www.sciencedirect.com/science/article/pii/S0166361521001615.

GOURIVEAU, R., MEDJAHER, K. e ZERHOUNI, N. (2016), *Health Assessment, Prognostics, and Remaining Useful Life*, cap. 3, 4 and 5, p. 33–136, John Wiley & Sons, Ltd, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119371052.ch5.

GUIGGIANI, M. (2019), in SPRINGER DORDRECHT, curatore, «The Science of Vehicle Dynamics - Handling, Braking, and Ride of Road and Race Cars», Second ed.

JEGADEESHWARAN, R. e SUGUMARAN, V. (2015), «Brake fault diagnosis using Clonal Selection Classification Algorithm (CSCA) – A statistical learning approach», *Enginee-*

ring Science and Technology, an International Journal, vol. 18 (1), p. 14–23, URL https://www.sciencedirect.com/science/article/pii/S221509861400055X.

JIN, X. e HAN, J. (2010), *K-Means Clustering*, p. 563–564, Springer US, Boston, MA, URL https://doi.org/10.1007/978-0-387-30164-8_425.

KHAN, S. e YAIRI, T. (2018), «A review on the application of deep learning in system health management», *Mechanical Systems and Signal Processing*, vol. 107, p. 241–265, URL https://www.sciencedirect.com/science/article/pii/S0888327017306064.

KLEPPMANN, M. (2017), in O'REILLY MEDIA, curatore, «Designing data-intensive applications : the big ideas behind reliable, scalable, and maintainable systems», First ed.

LI, X., DING, Q. e SUN, J.-Q. (2018), «Remaining useful life estimation in prognostics using deep convolution neural networks», *Reliability Engineering & System Safety*, vol. 172, p. 1–11, URL https://www.sciencedirect.com/science/article/pii/S0951832017307779.

MOBLEY, R. K. (2002), in «An Introduction to Predictive Maintenance (Second Edition)», Plant Engineering, Butterworth-Heinemann, Burlington, second edition ed., URL https://www.sciencedirect.com/science/article/pii/B9780750675314500130.

OMRAN, M., ENGELBRECHT, A. e SALMAN, A. (2007), «An overview of clustering methods», *Intell. Data Anal.*, vol. 11, p. 583–605.

PEPPES, N., ALEXAKIS, T., ADAMOPOULOU, E. e DEMESTICHAS, K. (2021), «Driving Behaviour Analysis Using Machine and Deep Learning Methods for Continuous Streams of Vehicular Data», *Sensors*, vol. 21 (14), URL https://www.mdpi.com/1424-8220/21/14/4704.

RAN, Y., ZHOU, X., LIN, P., WEN, Y. e DENG, R. (2019), «A Survey of Predictive Maintenance: Systems, Purposes and Approaches», URL https://arxiv.org/abs/1912.07383.

SANKAVARAM, C., PATTIPATI, B., KODALI, A., PATTIPATI, K., AZAM, M., KUMAR, S. e PECHT, M. (2009), «Model-based and data-driven prognosis of automotive and electronic systems», in «2009 IEEE International Conference on Automation Science and Engineering», p. 96–101.

SANKAVARAM, C., KODALI, A., PATTIPATI, K. R. e SINGH, S. (2015), «Incremental Classifiers for Data-Driven Fault Diagnosis Applied to Automotive Systems», *IEEE Access*, vol. 3, p. 407–419.

THEISSLER, A. (2017), «Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection», *Knowledge-Based Systems*, vol. 123, p. 163–173, URL https://www.sciencedirect.com/science/article/pii/S0950705117301077.

THEISSLER, A., PÉREZ-VELÁZQUEZ, J., KETTELGERDES, M. e ELGER, G. (2021), «Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry», *Reliability Engineering & System Safety*, vol. 215, p. 107 864, URL https://www.sciencedirect.com/science/article/pii/S0951832021003835.

WANG, G. e YIN, S. (2014), «Data-driven fault diagnosis for an automobile suspension system by using a clustering based method», *Journal of the Franklin Institute*, vol.

351 (6), p. 3231–3244, URL https://www.sciencedirect.com/science/article/pii/S0016003214000714.

WU, J.-D. e KUO, J.-M. (2010), «Fault conditions classification of automotive generator using an adaptive neuro-fuzzy inference system», *Expert Systems with Applications*, vol. 37 (12), p. 7901–7907, URL https://www.sciencedirect.com/science/article/pii/S0957417410003465.

ZHAO, R., YAN, R., CHEN, Z., MAO, K., WANG, P. e GAO, R. X. (2019), «Deep learning and its applications to machine health monitoring», *Mechanical Systems and Signal Processing*, vol. 115, p. 213–237, URL https://www.sciencedirect.com/science/article/pii/S0888327018303108.

ZHONG, J.-H., WONG, P. K. e YANG, Z.-X. (2018), «Fault diagnosis of rotating machinery based on multiple probabilistic classifiers», *Mechanical Systems and Signal Processing*, vol. 108, p. 99–114, URL https://www.sciencedirect.com/science/article/pii/S0888327018300657.

ZHOU, C. e PAFFENROTH, R. C. (2017), «Anomaly Detection with Robust Deep Autoencoders», in «Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining», KDD '17, p. 665–674, Association for Computing Machinery, New York, NY, USA, URL https://doi.org/10.1145/3097983.3098052.

Siti Web consultati

- Amazon Web Services – https://aws.amazon.com
- Towards Data Science, your home for data science – https://towardsdatascience.com
- PySpark Documentation – https://spark.apache.org/docs/latest/api/python
- Apache Parquet – https://parquet.apache.org
- Wikipedia – www.wikipedia.org

Ringraziamenti

Anche questo percorso è giunto al termine, definitivo questa volta. Emotivamente duro, mentalmente impegnativo, ma indubbiamente soddisfacente.

Ringrazio la mia famiglia, la mia ancora di salvezza nei momenti difficili. Non mi avete mai imposto di iniziare gli studi universitari e non mi avete mai obbligata a continuarli quando avrei voluto mollare tutto. Avete saputo gioire con me dei successi, ma anche supportarmi e, soprattutto, sopportarmi nei momenti di rabbia, di stanchezza e di paura, quando tutte le rinunce e gli sforzi fatti pesavano talmente tanto da opprimermi.

Con il vostro esempio mi avete plasmata, rendendomi chi sono oggi. Questo successo è anche il vostro. Un grazie, per voi, non sarà mai abbastanza.

Vi voglio bene.

Ringrazio il Professore Domenico Ursino, che con le sue lezioni è riuscito a trasmettere a tutti noi dei valori importanti, che mi porterò nel mio futuro lavorativo: l'impegno, la competenza e la passione per il proprio mestiere e ciò che lo riguarda.

A Deloitte, nelle figure di Giancarlo e Francesco e di tutti i ragazzi del Team, dico grazie per avermi subito accolta e guidata in questa nuova esperienza di vita.

Una menzione speciale è per Federica e Marianna, punti di riferimento imprescindibili per lo sviluppo dell'intero progetto, ma, soprattutto, fonti continue di nuovi spunti migliorativi e di conoscenza.

Ringrazio, infine, chi, in questi anni di studio, ha incrociato la sua strada con la mia, garantendomi un continuo supporto, tecnico e morale, tra lo sviluppo di un progetto e l'altro.

Concludo, augurando a me stessa che ciò che il futuro mi riserva sia ogni giorno migliore di quanto io possa, ora, immaginare.

Da qui inizia il mio nuovo viaggio.

Silvia