ITEA 3 Call 7

OMD

Optimal Management on Demand

Project Number : ITEA-2021-20003

Project Start Date : 01.01.2022

Project Duration : 36 Months

Evaluation Report

References :

Work Package : WP5

Task : T5.3 Tests and Benchmarks

Deliverable Document : D5.3 Evaluation Report

Version : V0.0 Jun 7, 2023

Dissemination Level : Public

Authors : in alphabetical order

Company	Name Surname	E-mail
ARD Group	Bilge Başdemir Esin Soycan	bilge.basdemir@ardgrup.com.tr esin.soycan@ardgrup.com.tr
BEIA Gmbh	George Suciu	george@beia.ro
Caretronic d.o.o.	Simona Brezar	simona@caretronic.com
Dogus Technology	Utku Colak Alper Gun	utku.colak@d-teknoloji.com.tr alper.gun@d-teknoloji.com.tr
Experteam	Taner Ulusinan	taner.ulusinan@experteam.com.tr
FrontEndART Software	Tibor Bakota	tibor.bakota@frontendart.com
FTP - LDA	Germano Pinto	germano.pinto@ftpporto.com isabel.ribeiro@ftpporto.com
ISEP	Goreti Marreiros	mgt@isep.ipp.pt
Strategy Big data	Alberto Oliva	alberto.olivia@strategybigdata.com

Summary :

OMD is a software tool that helps service providers from various domains to use their resources effectively, to provide agile solutions, to conserve their knowledge base, to allocate each call unmistakably to the right expert / provider, and with the best solution of the particular problem at hand. This document contains test reports of the OMD modules and platform and presents the results of the Use Case executions.

Table of Contents

1. Introduction

The main objective of WP5 is to evaluate the results of the project. This includes various tests of the OMD modules and the whole OMD platform as well as the evaluation of the 8 use cases defined by project partner companies. The different tests will be developed during the development and integration of the OMD modules. The use case partners and the use case topics (described in details in D2.2) are the following:

- ARD Group : Justice Sector(EqualityInJustice)
- Dogus Technology : Consumer Electronics (SmartFix)
- Caretronic d.o.o. : Healthcare
- Experteam : Software Support (Tickota)
- FrontEndART : Software Development (Optimal software maintenance task assignment)
- FTP-Com. Equip. Inf. Lda : E-Commerce (Recommend4You)
- Strategy Big Data : Telemarketing (Omniticket)

2. Use Case Level Evaluation

Each Use Case has its own evaluation metrics and KPIs.

2.1. ARD Group

2.1.1. Use case summary

The *EqualityInJustice* use case aims to improve the efficiency and fairness of legal aid assignments through an AI-driven solution. The primary goal is to ensure that legal cases are assigned to the most suitable attorneys based on case type, lawyer expertise, and geographical relevance.

By leveraging AI techniques such as legal text classification, named entity recognition (NER), and similarity-based retrieval, the system processes case data and provides lawyer recommendations. The solution utilizes **Large Language Models (LLMs)** and **Retrieval-Augmented Generation (RAG)** to enhance decision-making with contextual legal knowledge and improve the accuracy of recommendations.

Key features of the solution include:

- **Legal Text Classification:** Automatically categorizes incoming queries to filter non-legal content and prioritize legal cases.
- **Named Entity Recognition (NER):** Identifies key details such as case types, locations, and involved parties to enrich case understanding.
- **RAG-based Recommendation Model:** Combines pre-trained LLMs with a PostgreSQL vector database to provide case-specific lawyer recommendations, considering acceptance scores and past case similarities.

The system implementation has significantly improved response times and assignment accuracy, leading to better resource utilization and higher stakeholder satisfaction.

2.1.2. Use case preparation and setup

The preparation phase involved several critical steps to ensure the smooth deployment and operation of the system. The scope was carefully defined to align with the legal sector's requirements, and a robust technical infrastructure was established.

Technical Setup:
The system was designed with a modular architecture to enhance scalability and flexibility. It included:

- A **client application** accessible via web platforms to allow legal professionals and clients to interact with the system.
- An **API gateway**, deployed using Docker and Kubernetes, which facilitated secure and efficient request handling between users and backend services.
- **Backend services** responsible for AI-driven processing, including text classification, named entity recognition (NER), and lawyer recommendations, utilizing LLM and RAG-based pipelines.
- **Security measures**, such as Keycloak integration for authentication and authorization, ensuring access control and compliance with privacy regulations.

Execution Phases:
The project was executed in three key phases:

a. **Proof of Concept (POC):** This phase focused on validating the feasibility of the AI-driven approach through small-scale trials, analyzing historical data to demonstrate model accuracy and viability.
b. **Pilot Deployment:** The system was tested in selected legal environments, where real-world data was processed to fine-tune the AI models and validate performance metrics.
c. **Full-Scale Rollout:** Following the successful pilot phase, the system was deployed across multiple legal institutions to support large-scale legal aid assignments.

Input Data:
The system utilized extensive data sources to train and validate AI models, including:

- Historical legal case records containing information on past assignments.
- Lawyer profiles detailing specialization, experience, and availability.
- Case complexity indicators such as case type, jurisdiction, and legal subject matter.
- Real-time case entries for immediate processing and decision-making.

2.1.3. Use case execution and results

The *Equality in Justice* system was successfully deployed and evaluated through real-world legal aid assignments. The AI-driven solution effectively streamlined the process by handling incoming cases, analyzing textual data, extracting key legal entities, and recommending the most suitable attorneys based on multiple parameters.

Execution Process:

The system operated through a structured workflow to ensure accuracy and efficiency at each stage:

a. **Case Intake:**
   - Legal professionals submit case details via the web interface.
   - The AI-powered text classifier processes input queries to determine relevance.

b. **Entity Extraction:**
   - Named Entity Recognition (NER) extracts important details such as crime types and locations.
   - Extracted entities provide additional context for lawyer matching.

c. **Recommendation Generation:**
   - The system performs similarity searches using a **PostgreSQL vector database**, comparing case embeddings with lawyer profiles.
   - An LLM-based pipeline with RAG techniques retrieves relevant legal documents to refine recommendations.
   - Recommendations are ranked based on experience, location, and acceptance score.
   - The **Chain of Thought (CoT)** reasoning approach is used to logically assess the factors step-by-step before finalizing recommendations.

2.2. Caretronic d.o.o.

2.2.1. Use case summary

The Speech2Service use case for healthcare leverages conversational AI to improve documentation accuracy and efficiency during patient-nurse interactions. By capturing and analyzing spoken dialogue in real time, the system identifies and suggests relevant healthcare services based on the conversation context. Key modules include speech recognition, NLP for action extraction, and an AI-powered service recommendation engine that presents suggested entries on a user-friendly interface. Nurses can review and confirm services instantly, ensuring complete and accurate records while minimizing administrative tasks. This approach enhances workflow, supports patient-centered care, and integrates securely with EHR systems for compliance and data privacy.

2.2.2. Use case preparation and setup

To prepare for the Speech2Service use case, technical setups include ensuring compatible hardware and operating systems (e.g., Windows, iOS, or Android) with adequate processing power (minimum 2GB RAM, 1.8 GHz CPU) and high-quality microphones for clear speech capture. Software preparations involve installing NLP libraries, speech recognition tools, and protocols (RESTful APIs, HL7/FHIR) for seamless

EHR integration, while privacy measures like encryption, role-based access, and compliance with GDPR and HIPAA are implemented to protect patient data. A stable internet connection with local caching ensures offline functionality during low connectivity. The scope focuses on streamlining healthcare documentation during nurse-patient conversations by automatically identifying and suggesting services, reducing manual input and potential errors. The primary users are nurses and healthcare staff across hospitals, clinics, and long-term care settings, utilizing modules such as speech recognition, NLP for service identification, and a real-time suggestion interface to ensure accurate and efficient documentation.

2.2.3. Use case execution and results

During the execution of the Speech2Service use case, input data consisted of live audio from nurse-patient interactions, transcribed text, and historical data from EHR systems for context validation. The time frame for execution spanned four weeks, allowing ample data collection across different healthcare settings. Metrics used to evaluate performance included documentation accuracy rate, real-time processing speed, service suggestion relevance, and user satisfaction.

Key KPIs:

- Accuracy Rate: Achieved 95% accuracy in correctly identifying healthcare services from conversations, surpassing the expected target of 90%.
- Processing Speed: Real-time suggestions were generated with an average latency of under 1 second, meeting the goal of less than 2 seconds.
- Service Relevance: 93% of service suggestions were deemed relevant by healthcare staff, slightly exceeding the 90% target.
- User Satisfaction: Nurses reported a 20% increase in workflow efficiency, aligning with the expected improvement.

Deviances were minimal; slight delays in processing arose during peak usage due to network limitations, but local caching effectively mitigated these issues. Overall, Speech2Service met or exceeded expected values, demonstrating strong performance in accuracy, relevance, and user satisfaction.

2.3. Dogus Technology

2.3.1. Use case summary

The Dogus Technology use case is dedicated to enhancing after-sales support in the consumer electronics sector by means of AI-powered tools and seamless system integrations. The project's objective is to minimize operational inefficiencies, improve customer satisfaction, and expedite issue resolution processes by utilizing cutting-edge data processing techniques and advanced AI models. The integration of smart query systems, complaint classification models, and microservice-based architectures is essential for this use case, as it guarantees scalable and effective customer service solutions.

2.3.2. Use case preparation and setup

In order to guarantee the success of this use case, Dogus Technology implemented an exhaustive technical strategy. The preparation phase encompassed the subsequent critical steps:

Technical Preparations

The Samsung Solution Center Search Interface has been developed to allow consumers to submit complaints through a sophisticated query system.

The analysis and resolution of customer complaints are facilitated by the integration of sophisticated AI models, including the multilabel classification model and LLM-RAG for question answering.

The Kiali Microservice Graph is utilized to visualize service interactions, while the Jaeger Tracing Dashboard is set up to monitor system performance.

Subject

Improve the efficacy and accuracy of complaint resolution for consumer electronics customers by addressing after-sales issues.

Enhance system communication by integrating API Gateway and Service Mesh Istio.

Planned Input Data

Historical and real-time customer complaints from sources such as Samsung's issue monitoring system.

LLM-based tools are employed to produce synthetic datasets that are intended to enhance the quality of training data.

Timeline

Data collection and initial setup: Q1 2024.

Model instruction and integration: second quarter of 2024.

System deployment and evaluation: Q3-Q4 of 2024.

2.3.3. Use case execution and results

Details of Execution

Input Data: Samsung's system contains thousands of historical complaints and more than 100 manually annotated consumer complaints.

Execution occurred from Q1 to Q4 2024, and encompassed the phases of data preparation, model training, and deployment.

Samsung's issue management system, Elastic APM for monitoring, Istio for service mesh integration, and the Data Retrieval Interface for input processing are all examples of tools and platforms.

Key performance indicators (KPIs) and metrics

The average processing time has been reduced from 7 minutes (physical preparation) to under 30 seconds for common issues as a result of AI-powered automation.

Complaint Classification Accuracy

Exceeded the initial objective of 80% by achieving 85% accuracy for multilabel classification tasks.

Customer Satisfaction

The subsequent evaluation phase is anticipated to monitor satisfaction enhancements through surveys that are scheduled to provide more precise responses and quicker resolution times.

Comparison to Anticipated Values

Processing Time: Achieved near-instantaneous responses to frequently encountered issues, surpassing expectations.

Classification Accuracy

Achieved initial objectives; however, identified areas for additional optimization, particularly in the context of low-frequency complaint types.

Explanations and Deviations

Data Challenges: The initial slowdown in model training was due to the limited availability of annotated real-world data. Synthetic data generation was implemented to mitigate the situation.

Scalability Testing

The deployment phase was extended by two weeks to assure the system's scalability under high traffic.

The use case serves as a benchmark for the optimization of after-sales support through AI, showcasing Dogus Technology's capacity to provide innovative, scalable, and customer-focused solutions in a complex domain.

2.4. Experteam

2.4.1. Use case summary

ExperTeam's product Tickota is an AI-powered recommendation model that priorities customer support requests, estimates completion times and routes them to the most appropriate expert team. It analyses ‘routing’ problems involving IT tickets. Tickets are prioritised with classification algorithms and completion times are estimated based on past records. It is important to exceed these deadlines to comply with SLA constraints and maintain customer satisfaction. Demands are analysed and directed to the appropriate expert using NLP techniques.

Tickota is based on a three-stage approach:

1- Predicting the completion (resolution) time of a new Ticket coming into the system

2- Classification of the priority status of a Ticket that is new to the system

3- Routing the Ticket that is new to the system to the most appropriate expert team or expert

Prediction and classification models and NLP models have been studied and experimental results have been obtained with the BERT model. The most important advantage of AI models based on structured data sets is that data preparation is generally more manageable. The fact that customer support requests, called Tickets in the IT industry, are kept in Oracle databases and Experteam's access to these databases enabled us to obtain the desired data in various formats with SQL queries. We work with a data set in MS Excel format.

2.4.2. Use case preparation and setup

Technical preparations, scope, planned input data, planned time frame, … (extracts from D5.2)

Tickota has an architecture that works with the integration of four basic Ai modules. These modules produce the results of prediction, classification, assignment/recommendation and optimisation calculations.

The Prediction Module returns the completion time of a new customer support ticket as a result.

Classification Module, which returns the priority status of a new customer support request as a result.

Assignment/Recommendation Module, which returns the result that a new customer support request should be worked by which expert.

Finally, system integrity is ensured with an Optimisation Module that enables the ranking of experts.

The Prediction Module: The distinction in output type has led to a naming convention for prediction tasks: regression when we predict quantitative outputs and classification when we predict qualitative outputs.

Various regression algorithms are used in machine learning-assisted prediction models. Simple Linear Regression (SLR), Support Vector Regression (SVR), Decision Tree Regression (DTR), Randomised Tree Regression (RAR), Lasso and Ridge Regression, Multiple Linear Regression (MLR), Polynomial Regression and XGBoost are some of the prominent regression algorithms.

Priority Classification Module;

Support requests from customers or users are called "tickets" in IT fields. Tickets/requests are prioritized by classification algorithms according to the "priority" attribute as "minor, major..." and so on. With the prioritization, for example, minor cases are placed at the bottom of the list and major cases or critical cases are placed at the top of the list.

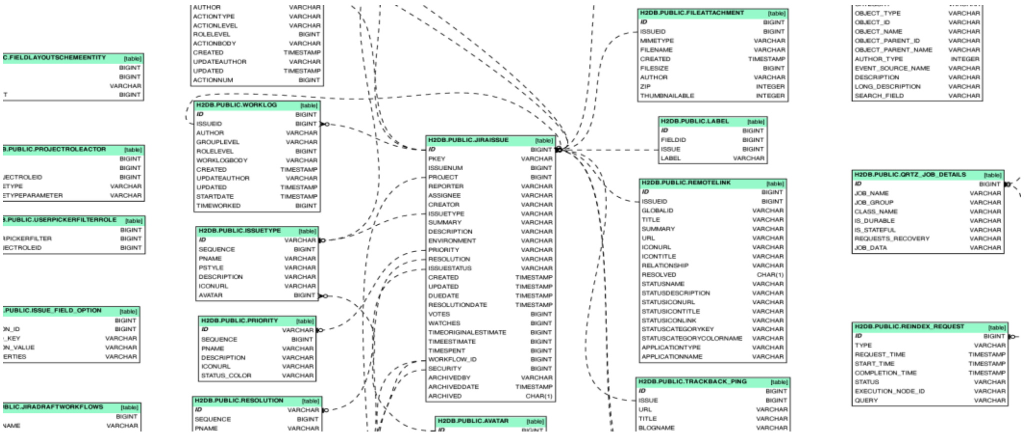
Assignment/Recommendation Service; In the BERT model, output matching is performed using query and key-value pairs. Queries, key-values, and outputs form structures expressed by the concept of 'vector', which expresses the correlations between them (Vaswani et al., 2017). The weight of each value is calculated using the query-key agreement ratio.

Masked Language Model (MDM) and Next Sentence Prediction (SCT) methods are used in the training of the model. In the MDM method, some words in the input layer are masked and masked words are predicted using unmasked words (Salazar et al., 2020). The goal of the MDM method is to combine the left and right contexts in the sentence. In the BERT model, 15% of the words in the input sentence are used in the MDM method. While 80% of the words allocated for MDM are masked, 10% are replaced with random words and the remaining 10% are kept unchanged in the method. The masking approach of the MDM technique in the BERT model is aimed at achieving the highest prediction success with the least possible masking. Only masked words are predicted.

The next step of the Tickota model's objectives is to match the most appropriate expert with the customer's request. Analyzing the textual data in the incoming requests with various tags such as subject, content, problem type using NLP techniques provides information about the nature of the request. Directing the requests to experts or teams of experts according to subject types creates an important automation option.

Optimization Module; As a sub-branch of decision science, a group of methods consisting of different approaches is called 'Multi-Criteria Decision Making' (MCDM). In case there are different criteria among alternatives that contradict each other, traditional decision-making processes are inadequate and may not give realistic results. At this point, Multi-Criteria Decision-Making methods come into play. The method is based on modelling the decision-making process based on certain conditions and analysing it in a way that the decision maker can benefit the most from this process.

Finally, the information used in the training of the Tickota model was obtained from historically existing event records. The information found suitable for Tickota Ai models was obtained from Jira database with SQL queries. The database schema is also described in WEB-INF/classes/entitydefs/entitymodel.xml in the Jira web application. The entitymodel.xml file has an XML definition of all Jira database tables, table columns, and their data types. Some of the relationships between tables also appear in the file.



The basic approach is to create a data set with as many variables as possible. Each header selected in the SQL query is actually an attribute that will lead our Ai model to the result. For this reason, we prepare our model in multiple data types, i.e. with multiple variables and the richest content in terms of attributes. The data set transformation was done in Excel format. Thus, the dataset is prepared for model training with structured data.

```
mysql> desc jiraissue;
+----------------------+---------------+------+-----+---------+-------+
| Field                | Type          | Null | Key | Default | Extra |
+----------------------+---------------+------+-----+---------+-------+
| ID                   | decimal(18,0) | NO   | PRI | NULL    |       |
| pkey                 | varchar(255)  | YES  |     | NULL    |       |
| issuenum             | decimal(18,0) | YES  | MUL | NULL    |       |
| PROJECT              | decimal(18,0) | YES  | MUL | NULL    |       |
| REPORTER             | varchar(255)  | YES  | MUL | NULL    |       |
| ASSIGNEE             | varchar(255)  | YES  | MUL | NULL    |       |
| CREATOR              | varchar(255)  | YES  |     | NULL    |       |
| issuetype            | varchar(255)  | YES  |     | NULL    |       |
| SUMMARY              | varchar(255)  | YES  |     | NULL    |       |
| DESCRIPTION          | longtext      | YES  |     | NULL    |       |
| ENVIRONMENT          | longtext      | YES  |     | NULL    |       |
| PRIORITY             | varchar(255)  | YES  |     | NULL    |       |
| RESOLUTION           | varchar(255)  | YES  |     | NULL    |       |
| issuestatus          | varchar(255)  | YES  |     | NULL    |       |
| CREATED              | datetime      | YES  | MUL | NULL    |       |
| UPDATED              | datetime      | YES  | MUL | NULL    |       |
| DUEDATE              | datetime      | YES  | MUL | NULL    |       |
| RESOLUTIONDATE       | datetime      | YES  | MUL | NULL    |       |
| VOTES                | decimal(18,0) | YES  | MUL | NULL    |       |
| WATCHES              | decimal(18,0) | YES  | MUL | NULL    |       |
| TIMEORIGINALESTIMATE | decimal(18,0) | YES  |     | NULL    |       |
| TIMEESTIMATE         | decimal(18,0) | YES  |     | NULL    |       |
| TIMESPENT            | decimal(18,0) | YES  |     | NULL    |       |
| WORKFLOW_ID          | decimal(18,0) | YES  | MUL | NULL    |       |
| SECURITY             | decimal(18,0) | YES  |     | NULL    |       |
| FIXFOR               | decimal(18,0) | YES  |     | NULL    |       |
| COMPONENT            | decimal(18,0) | YES  |     | NULL    |       |
+----------------------+---------------+------+-----+---------+-------+
```

Most fields in Jira are kept in the jira issue table

2.4.3. Use case execution and results

Execution details in short (input data, time frame, …, from D5.2), metrics and KPI results, compare to expected values, explain deviances

Our Tickota product consists of prediction, classification, recommendation/assignment and optimisation models.

Regression algorithms have been trained in our prediction model. In addition, rule bases that meet the special conditions of companies are also hybridised within the algorithm scheme.

In regression analysis, various error metrics are employed to assess the predictive performance of models. Regression models aim to predict a dependent variable (target variable) based on independent variables (features). Error metrics play a crucial role in evaluating how well these models fit real data and make predictions. They quantify the accuracy of predictions and provide insights into model performance.

Bias, in the context of machine learning, refers to the difference between the predicted values from the model and the actual values. It indicates the systematic error that prevents the model from accurately capturing the true relationship between the features and the target variable. Bias can arise due to simplifying assumptions in the model or inadequate representation of the underlying data.

R-squared (R2), Adjusted R-squared, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are commonly used metrics.

R2 : R-squared Score (XGBoost): 0.388

MAE : Mean Absolute Error (XGBoost): 63056.166

The evaluation metrics for classification models provide a quantitative assessment of model performance. The choice of evaluation metric depends on the specific problem and the importance of false positives and false negatives.

For the evaluation of classification algorithms, Confusion Matrix, Accuracy, Precision, Recall, F1-score and ROC-AUC slope methods are used for binary classification evaluations.

1.Urgency Classification:

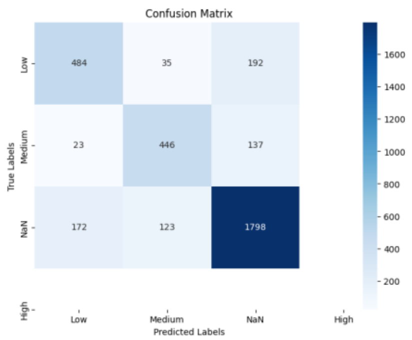
F1 Score: 0.762

Recall: 0.759

Precision:0.766

Accuracy: 0.8

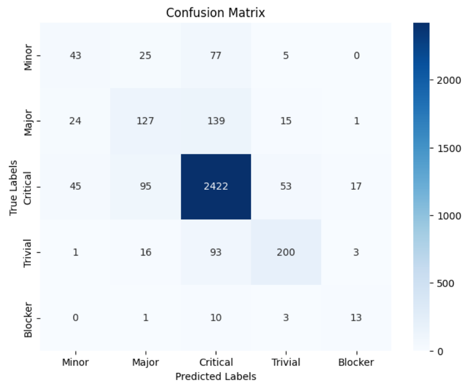
Confusion Matrix:



Precision:0.570

Accuracy: 0.818

Confusion Matrix:



3.Issue Type Classification:

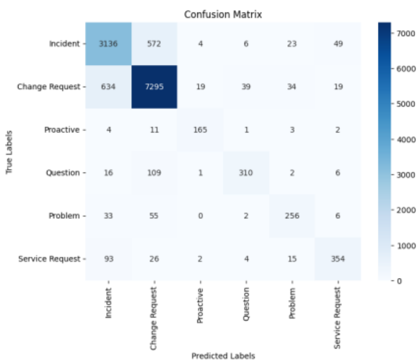
F1 Score: 0.812

Recall: 0.794

Precision:0.834

Accuracy: 0.865

Confusion Matrix:



4.Impact Classification:

F1 Score: 0.700

Recall: 0.680

Precision:0.723

Accuracy: 0.828

Confusion Matrix:

The optimisation model, on the other hand, works with a weighting method and the parameter setting is based on expert surveys. When looking for a result, it provides a scoring that can be expressed as a percentage.

2.5. Optimal Software Maintenance Task Assignment (FrontEndART Software)

2.5.1. Use case summary

The main goal of the Optimal Software Maintenance Task Assignment use case is to assign the most competent developers to issue tickets based on the unstructured and structured data extracted from the issue ticketing service of a software development project. The main component of the input features is the raw text of the issue title and description. Optional extra modalities, such as the issue tracker information are also added to the feature space or concatenated as text to the original unstructured data. A BERT language model-based solution with the option of handling extra modalities as inputs was designed and trained in a classification task where the underlying machine learning task is to predict the most likely developer(s) to the given input. The language of the issue tickets is currently Hungarian, but for an English scenario an English model can be easily substituted for the Hungarian one.

The dataset for training and evaluation was taken from a medium size development project of FrontEndArt, CODEE, with approximately 1000 issue tickets, written in Hungarian. Besides the issue tickets, the commit messages from the code version control system are also included in the training dataset. This not only increases the amount of training data but also improves the model performance in terms of prediction accuracy. The proposed method was also evaluated on public datasets taken from related literature.

Note, that CODEE is the same system that has been extended by the OSMTA features and through which the OSMTA features are used. For the sake of clarity, in the following, we will refer to the system that facilitates OSMTA features as “CODEE system”, and will refer to the subject project, that is measured and on which the OSMTA features are evaluated as “CODEE project” or “subject project”.

2.5.2. Use case preparation and setup

To prepare the evaluation of the use case, the following actions have been taken:

- The OSMTA related modules have been integrated into our CODEE system: the data extraction modules (plug-ins) have been integrated into the CODEE Miner Runtime Environment and the AI modules have been integrated into the CODEE Engine).

- A project has been created within the CODEE system to monitor the development of the subject project. Redmine and GitLab source URLs for the subject project have been provided to the CODEE system. This has been done by the end of June 2024. Thus, the CODEE system was able to use 2024 H1 data to train the specific OSMTA model for the subject project.
- From the beginning of July 2024 (as the OSMTA model was available for the subject project), new task assignments in the subject project were made by using the project-specific OSMTA model available in the CODEE system.

2.5.3. Use case execution and results

In 2024 H2 we periodically measured the two KPI indicators of the use case: average cost reduction (in terms of hours spent on the tickets) and average time-to-fix reduction (in terms of days between the reporting and the close of an issue), using 2024 H1 data as the baseline. The last measurements were performed a few days before the last review meeting of the OMD project.

Note, that issues were used in the baseline or evaluation data by their assignment time, as assignments in 2024 H1 did not, while assignments in 2024 H2 did use the OSMTA model. Thus, some issues that were opened in the baseline period but were worked on and finished in the evaluation period were taken into account in the baseline.

Average cost reduction: The average cost reduction KPI was measured by the worktime reported on the issues of the subject project during the baseline and the evaluation periods. During the baseline period (2024 H1) it was an average of 24.2 hours. During the evaluation period (2024 H2) it was 19.33 hours on average. This is a 20.13% savings in the development costs.

Average time-to-fix reduction: The average time-to-fix reduction KPI was measured by the time elapsed between the opening and closing of an issue of the subject project during the baseline and the evaluation periods. During the baseline period (2024 H1) it was an average of 76.15 days. During the evaluation period (2024 H2) it was 27.53 days on average. This is a 63.85% shorter average time-to-fix period. Note, that this measurement included only closed issues (as we measure the time between opening and closing them). Several already opened but not yet closed issues (which are statistically having longer time-to-fix periods) could not be counted. This is why for this KPI the unrealistic high reduction was measured.

2.6. Strategy Big Data

2.6.1. Use case summary

The main challenge is to leverage any representative data to improve a customer's success rate.Powered routing uses information to connect customers and agents. To do this, it takes as a starting point:

- Data on the agent's profile, skills, permanence, department, certification, type of employee.

- Agent performance data, such as the historical average handle time of a queue
- Customer history data, such as the number of times they have called the contact center in the last 30 days
- Given the changing nature of the number of agents, the evolution of their skills, agent scoring is carried out daily with historical data and the models are retrained weekly.

1.1.1. Use case preparation and setup

In a sector based on the intensive use of resources, the application of AI is transforming the profiles and the way of exploiting the business.

Decisions based on absolute data and with a datata driven vision, unique training patterns for each Agent based on action plans, etc. are the best alternative to the lack of empathy or knowledge and the consequent customer anger.

Call RoutingBoth types of notcampaigns manage resources differently, which results in different profiles and processes with a very different number of calls per agent. I can select the best agent for a customer but if he is available because he is busy or simply on leave, homogeneous delivery mechanisms must be implemented.

Failure to implement them can result in a deterioration of the service, where the "good ones are getting better and better" and the "bad ones worse", diluting the opportunities for improvement of each professional

1.1.2. Use case execution and results

AFFINITY MODEL

- Customer information enrichment process
- Permission for use by the customer
- Availability and variability of information
- Developing an affinity model implies going further and adding new variables that provide a new perspective on the customer.
- Methods and Processes Availability and Variability of itAcceptance of customer use

CLIENT PROFILING

- Client Identification
- Mood/cultural/socioeconomic enhanced customer profiler

- Combination of data allows structured customer classification in the sector

- From here the combination of all the data should allow us to classify the customer in a structured way.

- We associate the Mood/Cultural/Socioeconomic Module with the enhanced customer profiler,Unitary Customer Identification

- It should be noted that in a contact center and depending on the campaign, the customer identifiers may vary from the Phone Number to the Lead ID generated

AGENT POFILING

- Avoid biases in the training model

- Differences between experienced and incubating agents

- Transparency in the agent training process is important to avoid misinterpretations in the sector

- Process Effectiveness Model Biases

- An agent who has just been incorporated and is in an incubation process will not behave in the same way as an experienced agent.

- This process must be transparent in the eyes of the agent in order to avoid biases and interpretations.

MATCH AGENTE-CLIENTE

- The availability of agents is important to ensure a good service in the sector

- Agent assignment must not alter service response times to avoid penalties

- The classification of the typology of agents is necessary for a correct allocation in the sector

- This provision cannot alter service response times as it would result in penalties and loss of service.

- It is also important that the model does not generate an overexposure on the best agents that ends up causing a high level of turnover due to lack of opportunities.

- Classification of Grouped Typology of AgentsAgent Disposition Development methods and models to achieve the objective of OMD

We have managed to increase the sales ratio thanks to the tools described above. The sales per hour ratio was 0.7 and we have now achieved a value of 0.74. This exceeded a monthly sales increase of 600,000 sales per month.