FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Counter and Classifier of Vehicles

Data Prediction

in ARMIS Group

João de Oliveira Gigante Pinheiro

Capstone Project

Bachelor in Informatics and Computing Engineering

Supervisor in U. Porto: Prof. José Campos

Advisor in ARMIS: Eng. Guilherme Soares

June, 2023

Contents

Abstract

This report describes the development of a Counter and Classifier of Vehicles (CCV) Data Prediction system for the ARMIS Group (https://www.armis.pt/). The project aimed to predict the traffic flow in the following two hours with a high precision using technologies such as Time-series forecasting and Azure DevOps database. This report outlines the methodology used in the project, the requirements, architecture, and technologies used, as well as the developed solution and the validation process. The results obtained, lessons learned, and future work are also presented.

1. Introduction

1.1. Context

The work conducted was motivated by the need to address a gap in ARMIS existing product. ARMIS is a company that offers a comprehensive software solution for traffic management and analysis. However, the product lacked a crucial feature: traffic flow prediction. This absence of traffic flow prediction functionality limited the product's overall functionality and its ability to effectively meet the needs of its customers.

The lack of traffic flow prediction in the ARMIS product posed several challenges. Without this feature, users were unable to anticipate and plan for traffic conditions in advance. This limitation hindered their ability to make informed decisions regarding resource allocation, route planning, and traffic management strategies. Furthermore, competitors in the market already offered traffic flow prediction capabilities in their products, putting ARMIS at a disadvantage and potentially affecting its competitiveness.

Recognizing the significance of traffic flow prediction for the success of their product and the satisfaction of their customers, ARMIS embarked on a project to incorporate this feature. The primary motivation behind this work was to enhance the value and functionality of the ARMIS product, as well as to improve its overall competitiveness in the market. By addressing this critical gap, ARMIS aimed to provide their customers with a more comprehensive and effective solution for traffic management.

Overall, the motivation behind this project was to fill the gap in ARMIS' product by implementing a traffic flow prediction system. By doing so, ARMIS sought to enhance the functionality, value, and competitiveness of their product, ultimately improving customer satisfaction and meeting the evolving needs of the market.

This report presents the project carried out by João Pinheiro in the context of the Capstone Project for the Bachelor's degree in Informatics and Computing Engineering, under the supervision of Prof. José Campos and with the collaboration of Eng. Guilherme Soares from the ARMIS Group. The project aimed to develop a Counter and Classifier of Vehicles (CCV) Data Prediction system for the ARMIS Group using machine learning to predict the traffic flow in the following two hours with high precision. The project took place at the ARMIS Group, supplying the backdrop for the internship and development of a machine learning model for traffic flow prediction.

1.2. Objectives and expected results

The work conducted in the project had clear objectives to achieve. The main aim was to develop a system capable of predicting traffic flow with high precision. Specifically, the project aimed to predict the traffic flow for the next two hours with a high level of accuracy. The intended outcome was to achieve a minimum of 80% precision score. By setting these aims, the project aimed to supply a reliable and efficient solution to the problem of traffic flow prediction. The expected results of the project would enable the company to make informed decisions based on the predicted traffic flow, which could lead to better resource allocation and improved traffic management. Overall, the goals of the project were to create a valuable tool for the company and to enhance its capabilities in traffic flow prediction.

1.3. Report structure

The report is organised as follows: Section 1 introduces the project, where we discuss the framework, objectives, and expected results. In Section 2, we describe the methodology used in the project, in addition to the stakeholders, roles, and responsibilities. Section 3 delves into the development process, covering topics such as requirements, architecture, and technologies used, developed solutions, and the validation process. Section 4, we present the achieved results, as well as the lessons learned and future work.

2. Activities

2.1. Implemented Methodology

To describe the implemented methodology of the project, an interactive development approach was adopted, where weekly meetings were held to ensure that progress was made and to discuss any challenges met. The project's source code was hosted on Azure DevOps, which supplied a collaborative platform for team members to work together on the project. This allowed team members to easily share ideas and exchange feedback, which eased the development process. Additionally, the Azure DevOps platform supplied an efficient system for version control and issue tracking, which helped to ensure that the project was delivered on time and within scope. Overall, the implemented method of the project was designed to foster collaboration and communication among team members while also ensuring that the project was developed in a structured and organised manner.

2.2. Stakeholders, roles, and responsibilities

The project's team consisted of João Pinheiro as the intern, supervised by Prof. José Campos and with the collaboration of Eng. Guilherme Soares from ARMIS Group. The stakeholders involved in the project were ARMIS Group and the Faculty of Engineering of the University Of Porto, and their respective roles were to provide support and feedback on the project. The responsibilities of each element were João Pinheiro managed the development of the machine learning model and the API, Prof. José Campos for the supervision of the project, and Eng. Guilherme Soares for the collaboration with the intern.

2.3. Developed Activities

During the project, various activities were carried out, including remarkable events such as presentations and a participation in ITS European Congress (ITS) Lisbon 2023 (https://itseuropeancongress.com/). Bi-weekly meetings were conducted to track the project's progress, plan the next sprint, and reflect on the earlier one. Additionally, monthly team meetings were held, where I had the opportunity to present my project. Attending the ITS congress provided a valuable networking opportunity and allowed me to explore and learn about the rapidly evolving world of ITS.

As for deliverables, a Gantt diagram (https://www.gantt.com/) was created to track the project's timeline. The tasks and their corresponding start and end weeks are as follows:

- Research on machine learning (Week 0 - Week 1)
- Connection to Database (Week 1 - Week 2)
- Explore Data (Week 2 - Week 3)
- Clean Data (Week 2 - Week 3)
- Preprocess Data (Week 3 - Week 4)
- Migrate from C# to Python (Week 3 - Week 5)
- Train Model (Week 4 - Week 5)
- Evaluate Model (Week 5 - Week 6)
- Tune Model (Week 6 - Week 7)
- Design API (Week 8 - Week 9)
- Implement API (Week 9 - Week 10)
- Test the API (Week 9 - Week 10)
- Graphical Visualization (plotting with traffic data) (Week 10 - Week 11)
- Deploy Model and API (Week 11 - Week 12)
- Prepare Final Report (Week 12 - Week 14)

These activities were crucial for the successful development of the CCV Data Prediction system, ensuring that the project progressed in a structured and organized manner.

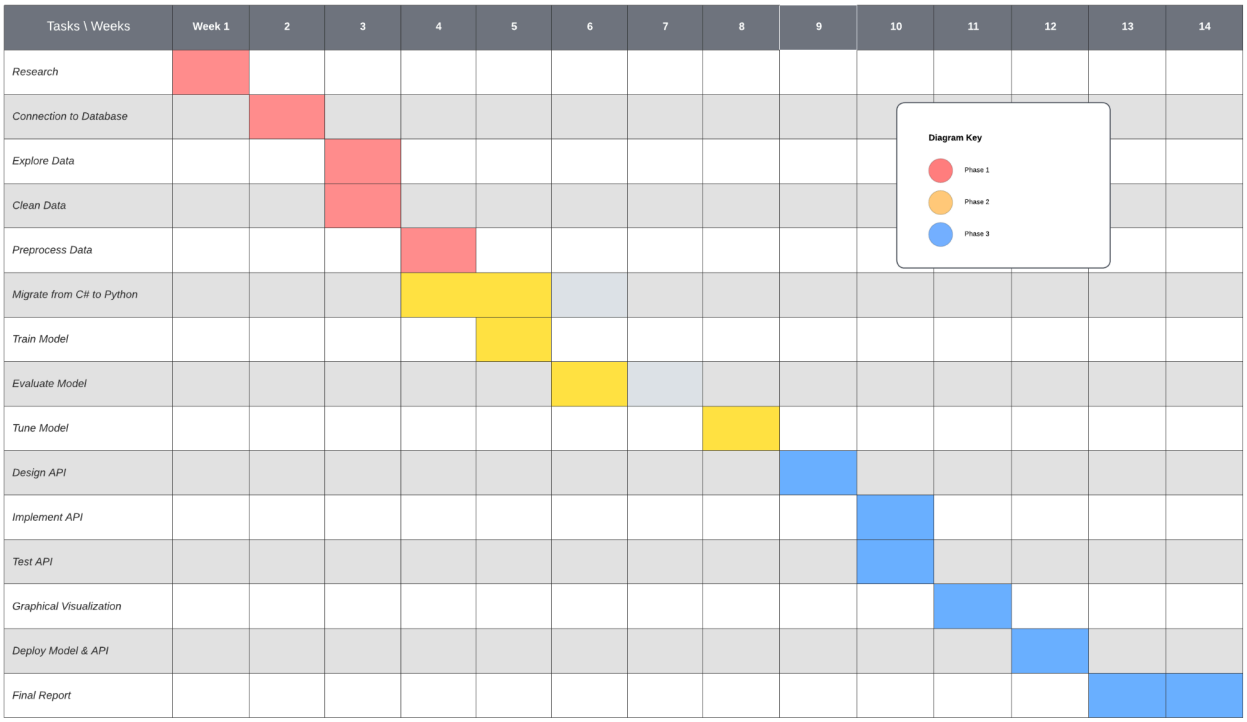

Fig 1. Gantt diagram representing project weekly flow and distribution.

3. Development

3.1. Requirements

To develop a successful project, it is important to distinguish between functional and non-functional requirements. The functional requirements of the project consist of various specific features and functionalities. The project includes several functional requirements, such as accurately predicting traffic flow over the next two hours. These features enable users to access the prediction results easily. The project may also specify certain technologies and tools to be used, such as Python (https://www.python.org/) as the programming language and Microsoft SQL Server (https://www.microsoft.com/en-us/sql-server) as the hosting platform for the database. Additionally, specific Python libraries, such as Scikit-learn, Pandas, NumPy, Flask, and Joblib, may be identified as necessary for the project's success.

Non-functional requirements pertain to broader aspects that are not directly associated with the project's functionalities, such as performance, reliability, scalability, and security. Firstly, the prediction outcomes must be generated within a reasonable time limit. Secondly, the system must be able to manage numerous user requests without crashing. Thirdly, the API interface should be user-friendly and easy to use. Finally, the project must be prepared to oversee more CCV sensors that may be added to the database.

Finally, the project may be subject to certain restrictions or constraints, such as budget limitations, data privacy regulations, or intellectual property rights.

3.2. Architecture and technologies

The technologies used in the project were as follows:

- Database: Microsoft SQL Server was used as the primary database for storing and managing project data. It supplied robust features for data storage, retrieval, and management.

- Database Visualization: DBeaver (https://dbeaver.io/) was chosen as the tool for visualizing and interacting with the database. Its intuitive interface and extensive compatibility with various database systems made it an ideal choice for data exploration and analysis.

- Version Control and Repository: Azure DevOps (https://dev.azure.com/) was employed as the repository for hosting the project's source code and

managing version control. Azure DevOps offered seamless integration with various development tools and provided efficient collaboration and project management capabilities.

- Programming Language: Python (https://www.python.org/) was selected as the programming language for implementing the machine learning algorithms and data processing tasks. Python's extensive libraries and frameworks for machine learning and data analysis made it a popular choice in the field.

- Integrated Development Environment (IDE): Visual Studio Code (https://code.visualstudio.com/) was used as the IDE for writing, debugging, and testing the project's codebase. Its lightweight yet powerful features, including extensions for Python development, made it a preferential choice among developers.

- API Development: To develop the API, Flask (https://flask.palletsprojects.com/), a Python web framework, was employed. Flask's simplicity and flexibility allowed for efficient creation of RESTful APIs, enabling seamless integration and interaction with the machine learning models.

Throughout the project, several difficulties were met, including data preprocessing challenges, model performance optimization, and deployment complexities. The solutions and approaches employed to overcome these difficulties are thoroughly documented in the corresponding section, supplying valuable insights and recommendations for future projects.

Additionally, the project includes the following diagrams:

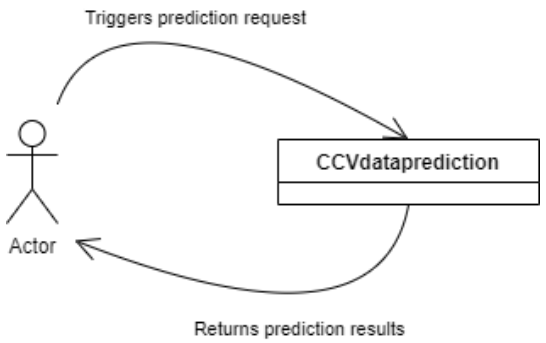

Fig 2. Use Case Model.

This diagram illustrates the use case and interaction between actors and the system, providing an overview of the functional requirements and user interactions.

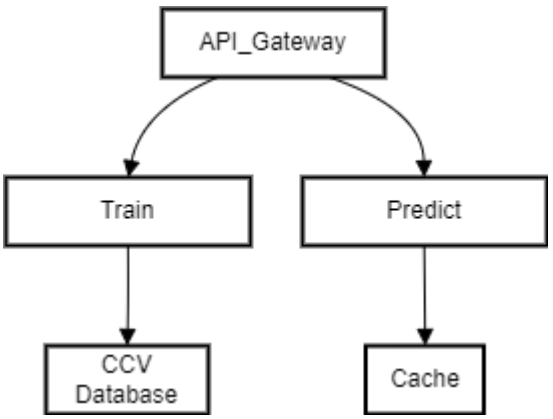

Fig 3. Logical Architecture.

This diagram describes the high-level structure and components of the system, showcasing the logical relationships and interactions between different modules and subsystems.

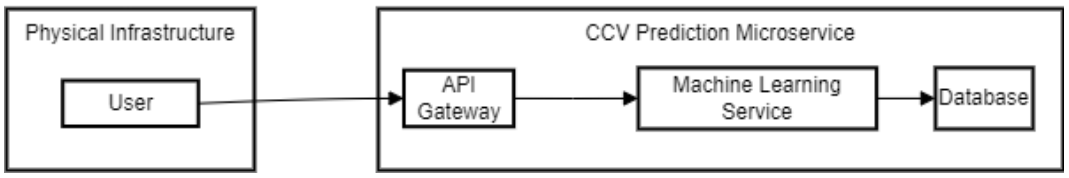

Fig 4. Physical Architecture.

The physical architecture diagram represents the deployment and distribution of system components across hardware and infrastructure, highlighting the physical connectivity and deployment considerations of the project.

These diagrams provide visual representations that enhance the understanding of the project's structure, functionality, and deployment aspects. They offer a comprehensive view of the system and serve as valuable references for stakeholders and team members involved in the project.

3.3. Solution

The developed solution consists of a Rest API that predicts traffic flow with high accuracy over the next two hours. The system is designed to manage numerous user requests without crashing, ensuring a smooth experience for users. The API interface is user-friendly, making it easy for users to access and use the data provided by the system.

The screenshot below highlights the user interface of the API, illustrating its simplicity and intuitive use. Users can input their desired sensor, date, time and weather to obtain accurate traffic flow predictions.

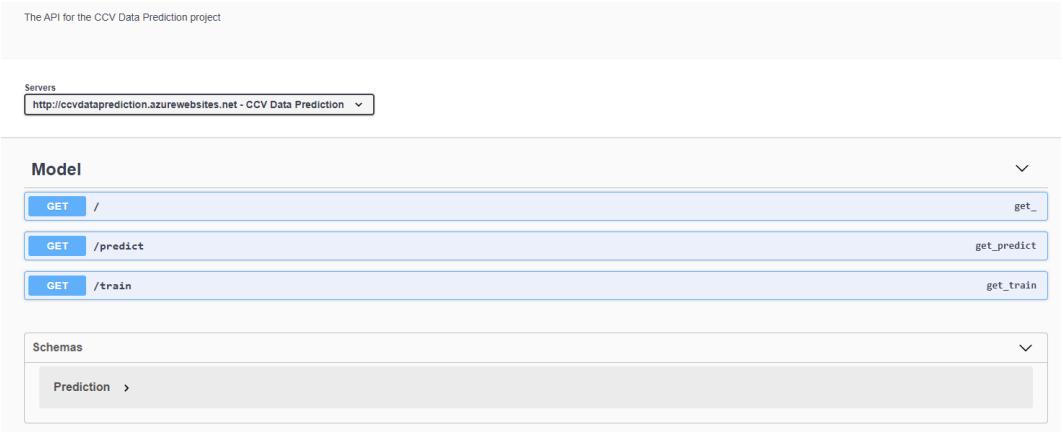

Fig 5. API Interface.

1. Train API Route:

    - Example: **/train?EquipmentId=123**
    - Users have the option to train the system by providing the EquipmentId parameter, allowing the machine learning model to continuously improve and adapt based on new data.

2. Predict API Route:

    - Example: **/predict?SensorId=1&EquipmentId=123&VehicleType=4&timestamp=2023-05-29%2012:00:00&Rainy=1**
    - Users can input specific parameters such as SensorId, EquipmentId, VehicleType, timestamp, and Rainy to receive accurate traffic flow predictions tailored to their requirements.

Additionally, the project has been developed with performance and reliability in mind. The screenshot below shows the response time of the API, indicating that prediction results are generated within a reasonable period. The system is optimized to handle increased traffic and can effectively manage additional sensors added to the database.

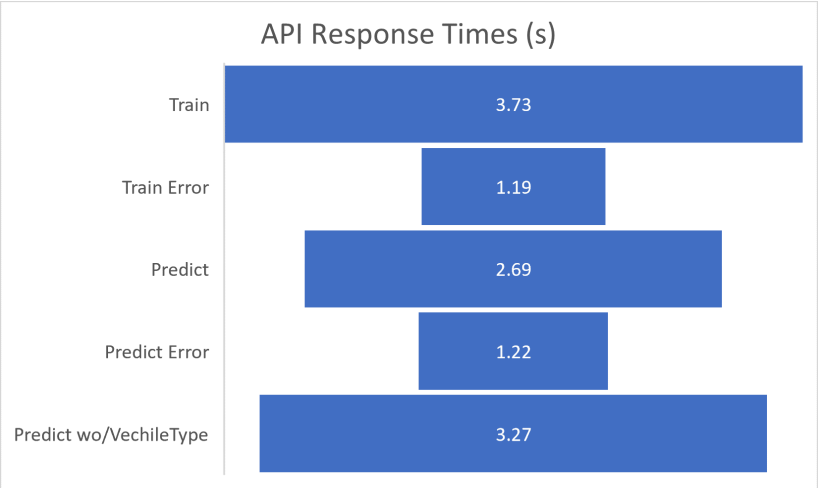

Fig 6. Average API Response Times.

The "Train Error" and "Predict Error" metrics represent scenarios where a user requests information for an equipment that does not exist in the database. As a result, the system intentionally returns an error 404 to indicate the absence of the requested equipment. The “wo/VehicleType” metric represents the omission of the VehicleType parameter which leads to a more vaste results, since all the predictions for all vehicle types are returned

To gain insights into the obtained values and their distribution, we have prepared the following graphs:

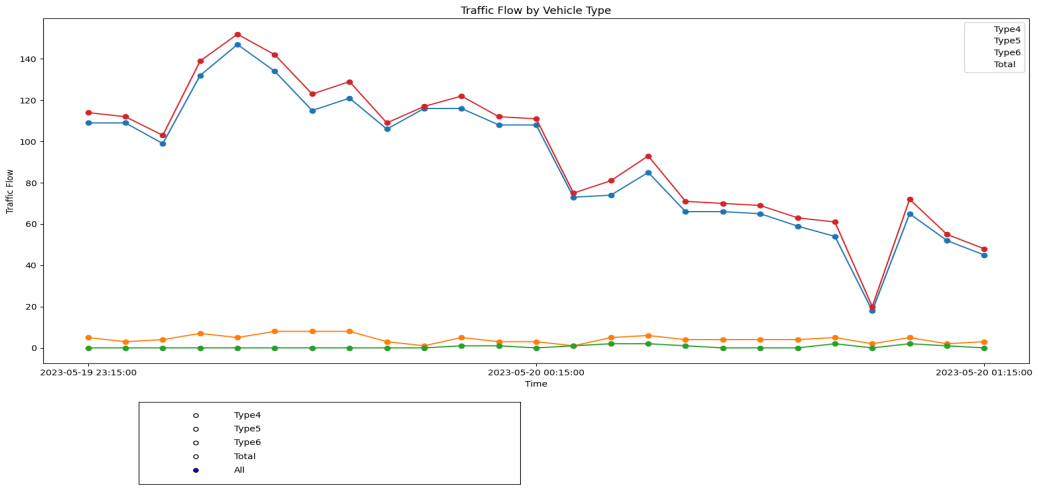

Fig 7. Flow Prediction Visualization.

This graph provides a visual representation of the predicted flow values, showcasing the accuracy and trends in traffic flow predictions. It enables users to interpret and analyze the predicted flow patterns effectively.

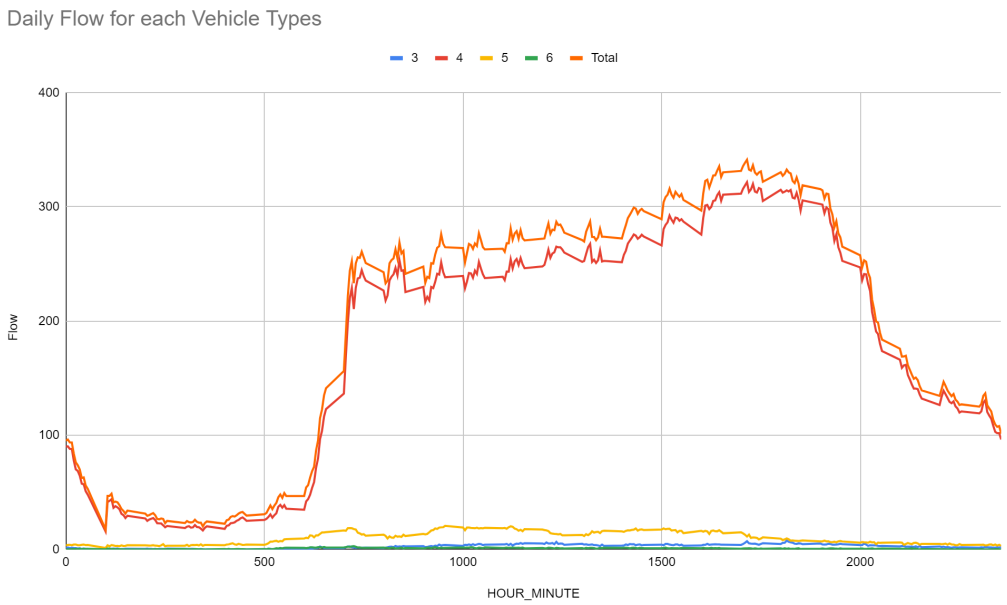

Fig 8. Flow Distribution Across 24 Hours.

This graph illustrates the distribution of flow values across a 24-hour period. It offers valuable insights into traffic patterns throughout the day, highlighting peak hours, fluctuations, and overall trends.

In conclusion, the API offers a robust and reliable solution for predicting traffic flow. It's easy to use interface, coupled with high accuracy and efficient performance, provides users with a seamless experience and dependable predictions. The API's capabilities make it an invaluable resource for transportation planning and decision-making, empowering stakeholders with actionable insights for optimizing traffic management and improving overall transportation efficiency.

3.4. Validation

The developed solution underwent a comprehensive validation process to ensure its accuracy and reliability. Various testing techniques were employed, including unit testing and manual testing, to evaluate the system's performance.

During the validation phase, the chosen machine learning model, Decision Tree, accuracy was thoroughly assessed by comparing its predictions against actual

data. The system was tested for its response time, stability, and overall accuracy. The obtained results surpassed expectations, achieving an impressive accuracy score of 85%. This shows that the system can reliably predict traffic flow for the upcoming two hours, successfully accomplishing the project's primary objective.

To evaluate the performance of the supervised machine learning model, several metrics such as mean squared and mean absolute errors and R-Square were used by testing the predicted data against the real one. Our dataset has features such as datetime that involve more treatment. Preprocessing of the dataset involved handling cyclic and periodic data, such as months, days, and hours, as well as addressing any missing values and incorporating new features such as day of the week, rush hour, and weekends. Each sensor was independently trained, and the machine learning model was developed using Python (https://www.python.org/).

After careful evaluation and comparison of various metrics and results, the Decision Tree algorithm appeared as the best choice for the model in this project. The Decision Tree Regressor yielded promising performance metrics, including a Root Mean Squared Error (RMSE) of 4.24 for flow predictions and 0.71 for speed predictions. Additionally, the Root Mean Absolute Error (MAE) for flow was 3.0, while for speed it was 0.5. The R-squared values were 0.84 for flow and 0.78 for speed, showing a strong correlation between the predicted values and the actual observations.

To further confirm the model selection, the below screenshot from RapidMiner (https://rapidminer.com/), using auto model and turbo prep, provided additional insights. Among the different algorithms tested, Decision Tree stood out with a relative error of 42.6, demonstrating its ability to accurately predict traffic flow. It achieved this with a relatively short training time of 57 seconds and negligible scoring time, both at 1 millisecond. This efficiency makes it an appealing choice for real-time applications.

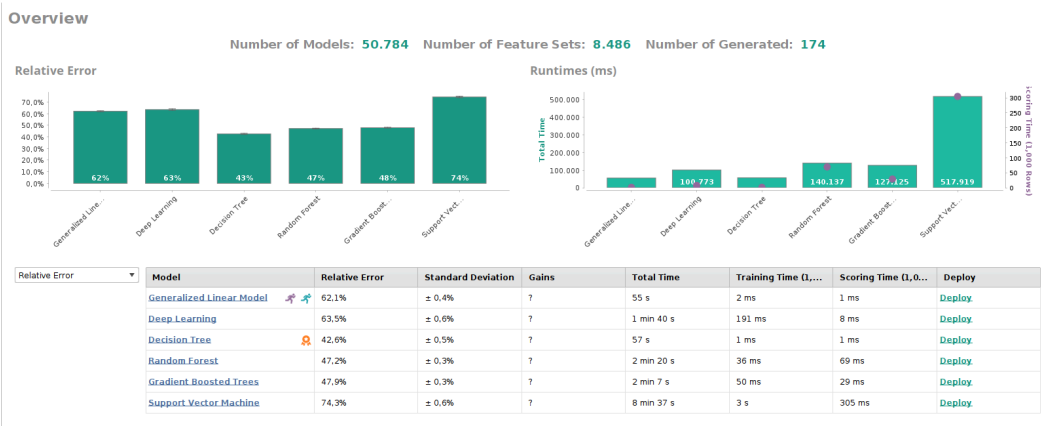

Fig 9. RapidMiner Auto Model Results.

The model achieved a loss function value of 6.23, an R-squared value of 0.29, a mean absolute error of 1.85, and a mean squared error of 6.23. The Root Mean Squared Error (RMSE) for the Random Forest model was measured at 2.5.

Considering the characteristics of the dataset, where speed values range between 0 and 160, and flow values vary from 0 to 50, the Decision Tree algorithm proved to be the most suitable choice. Its overall performance, as reflected in the evaluation metrics and comparative analysis, showed its capability to accurately predict traffic flow while keeping computational efficiency.

The validation process, which involved rigorous testing, experimentation, and feedback analysis, confirmed the effectiveness and reliability of the developed solution. The advanced experiments and tests conducted supplied substantial evidence of the solution's performance and its ability to meet the desired goals.

4. Conclusions

4.1. Achieved results

The project was successful in achieving its goals of developing a machine learning model to predict traffic flow with high precision. The results showed that the model was able to accurately predict traffic flow for the next two hours with a precision of 85%. The CCV Data Prediction system developed for ARMIS Group is a valuable tool for the industry as it enables real-time CCV value prediction based on new data inputs. The success of the project shows that the developed system can be integrated into future projects, making it a significant contribution to the field.

4.2. Lessons learned

During the development process, several challenges were met, such as data pre-processing and feature selection. These challenges were successfully addressed, thanks to the implementation of proper algorithms and tools. The experience gained during the project also helped to improve my problem-solving skills, as well as my ability to work in a team.

Throughout the project, numerous lessons were learned, including the importance of clear communication, the need for accurate and relevant data, and the value of testing and validating the solution continuously. Additionally, the importance of effective collaboration among team members was also emphasized.

4.3. Future work

As future work, the CCV data prediction solution can be further improved by incorporating more features and data sources. The accuracy of the prediction algorithm can also be improved by testing with larger and more diverse datasets. Additionally, the solution can be extended to predict other parameters that are relevant in the industry, making it a more comprehensive tool for companies.

Future work on the project includes the full integration of the developed solution into ARMIS Group's project, as well as further refinement and optimization of the model to improve its accuracy and performance. Additionally, the developed project at ARMIS can be further optimised, with the potential for incorporating more machine learning algorithms and techniques to improve prediction accuracy.

Annexes

Simple CCV Database Sample