Enhancing Automotive Service Efficiency with Predictive Time Forecasting

Jesso Joseph
PG Scholar
Department of Computer Applications
Amal Jyothi College of Engineering
Kanjirappally, Kerala
jessojoseph24@gmail.com

Sr. Elsin C. SH
Assistant Professor
Department of Computer Applications
Amal Jyothi College of Engineering
Kanjirappally, Kerala
elsinchakkalackal@amaljyothi.ac.in

***Abstract*— In today's fast-paced world, optimizing the automotive industry is of utmost importance. This conference paper presents a novel method for optimizing vehicle operations using predictive forecast models. These models can accurately estimate maintenance time for service providers, reducing uncertainty in maintenance schedules. Using advanced machine learning techniques, the predictive models analyze various factors such as vehicle type, service history, maintenance need to forecast service time and guidelines for decision-making processes. By simplifying decision-making for service providers, this approach is not that time-saving not only improves overall vehicle operation performance This article describes in detail the methodology, data sources and key components of a weather forecast model. As the automotive manufacturing industry continues to evolve, our model represents an important step towards optimizing operations and improving customer satisfaction.**

Keywords— Random Forest, Machine Learning.

I. INTRODUCTION

Improving operational efficiency and accuracy is essential to meet the demands of a dynamic and rapidly changing industry in automotive performance management. This symposium offers a revolutionary approach to the automotive service environment. It combines weather forecasting models with principles of machine learning and data analytics to redefine service delivery and improve overall service delivery. The main objective of this model is to provide vehicle service providers, obtain accurate forecasts of the time required for various service tasks In addition to this invaluable information, the model goes further by providing personalized recommendations designed to streamline and optimize the entire service process.

At its core, this new approach uses advanced machine learning techniques, including random forest regression, which are known for their accuracy and scalability. These models are designed to accommodate the variations of their users have defined and determined the optimal timing of vehicle maintenance schedules. These variables include basic factors such as vehicle type, service history and specific maintenance needs as well as more complex details such as service history and maintenance requirements, but the sample size goes further. It seamlessly integrates important vehicle information such as past maintenance, mileage and others to ensure comprehensive analysis for service providers and owners who need accurate information about maintenance activities.

In summary, this seminar serves essential purposes in the automotive service industry. It adds significant value by saving time and making the vehicle maintenance process more efficient. This article delves deeper into the complex workings of the model, providing a comprehensive understanding of the methods, data sources, and key components of the model. Ultimately, it shines as a beacon of innovation in the ever-changing auto services landscape.

II. RELATED WORKS

Ouhilal Merriam and team [1] compared forecasting algorithms such as Linear Regression, Support Vector Machines, and Multilevel Perceptron for time series forecasting. Given a baseline comparison, the limited experimental design makes it difficult to conclude definitively that linear regression is the most accurate algorithm The results suggest the need for further research using a more robust methodology before selecting the optimal time series forecast method.

Traub and the team at BMW [2] proposed changes in vehicle electrical and software design, including hierarchical E/E architecture, central communications servers, and service-oriented architectures. They thought to drive consumer IT trends automotive industry. However, conclusions about potential benefits are strongly influenced by the intent of architectural proposals.

Cornelius [5] proposed various methods to estimate the available processing time per unit, including classification of lost time and informal analysis. He considers things like meetings, breaks, vacations, and sick leave that reduce hours. However, the non-randomization of the surveys and the limited data provided significantly affect the conclusions drawn in determining the available working hours.

Owens and Luh [4] proposed various methods to estimate task completion time under uncertainty, including stochastic estimation of future task arrivals. Model problems were tested with weighted delay objectives. However, the selection of problems and the limited computational analysis greatly influence the conclusions on the merits of the proposed methods.

III. METHODOLOGY

A. Data Loading and Preprocessing:

The method begins with data loading and preprocessing. The code starts by introducing libraries needed for data manipulation and machine learning, such as pandas for data processing, and Scikit-learn (Sklearn) for the machine learning module. The dataset is loaded from the CSV file 'bike_service_data.csv' using the pd.read_csv function. Categorical features are carefully labeled with encoders, and the focus may vary depending on the specific characteristics of your vehicle service data set. These encoding changes are important in preparing data for analysis and modeling. Then, the data set is logically partitioned into two parts: attribute scope (X) and target variable (y). The primary purpose of this section is to determine the vehicle's maintenance "service duration" based on various components of the data set. To facilitate model analysis, the dataset is intelligently divided into training data (80%) and test data (20%).

B. Model initialization and training:

The chosen model for this methodology is a random forest regressor. It is initialized with vital parameters, including the number of estimators (100) and a random seed (42) for result reproducibility. The model is then trained using the preprocessed data. During this phase, the random forest regressor is applied to the feature set (X) and the target variable (y). The trained model plays a pivotal role in predicting the service duration for vehicle maintenance.

C. Model and Label Encoder Saving:

Upon successful training and evaluation, the model is saved to a file named 'rf_model.pkl' using the 'joblib.dump' method. This step ensures that the trained model can be readily applied in future applications without the necessity of retraining. Furthermore, the label encoders employed for data preprocessing are also preserved in separate files, guaranteeing consistent data encoding in future use cases. Each label encoder is stored with a descriptive filename for easy reference.

D. User Input and Prediction:

To forecast the service duration for a vehicle, the code interacts with the user to collect specific information about the vehicle in question. This information encompasses vehicle model, age, service type, service history, build year, mileage, last serviced date, and the current year. The user's input is structured into a DataFrame for processing.

E. Predictions and Property Analysis:

Based on the user's input, the model is utilized to predict the service duration for the vehicle. Predictions are established on patterns acquired from the training data. Additionally, the code incorporates a vehicle analysis component. If the projected service time exceeds a predefined threshold (currently set at a specific duration), the system offers personalized recommendations to the vehicle owner. These recommendations encompass various aspects, including maintenance suggestions, service history insights, and guidance to improve vehicle performance and longevity.

F. Model Evaluation and Accuracy Metrics:

Subsequently, the model's performance is meticulously assessed using key accuracy metrics. The Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared score (R2) are calculated. These metrics provide valuable insights into the predictive capacity of the model and the quality of its predictions. They serve as benchmarks for measuring the model's effectiveness.

G. Model Accuracy Metrics and Property Tips Display:

Finally, the seminar paper offers a comprehensive assessment of the model's accuracy. The accuracy metrics, including MAE, MSE, RMSE, and R-squared (R2) score, furnish critical insights into the model's alignment with actual data and its predictive accuracy. It considers factors like vehicle age, service type, service history, and maintenance recommendations. These insights are instrumental in understanding the model's reliability in the context of vehicle service duration predictions, aiding vehicle owners in making informed decisions to optimize their vehicle maintenance processes.

IV. BUILD MODEL

A. To construct a robust and precise machine learning model for effective service duration prediction, we adhere to the subsequent procedures:

B. Importing Libraries

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
import joblib
```

a) pandas:

A Python data manipulation and analysis library that provides data structures and functions for structured data processing.

b) RandomForestRegressor from sklearn.ensemble:

Machine learning algorithms used for regression tasks that leverage a collection of decision trees to predict numbers.

c) LabelEncoder from sklearn.preprocessing:

A tool that encodes categorical features into numbers that make suitable for machine learning models.

d) mean_absolute_error,mean_squared_error,

e) r2_score from sklearn.metrics:

Functions for assessing regression model performance, including metrics like mean absolute error, mean squared error, and R-squared score.

f) joblib:

A Python library tailored for lightweight pipelining, specifically designed for parallel processing and efficient data sharing.

C. Building the model

1) Loading the Dataset:

The code begins by loading a dataset from a CSV file named 'bike_service_dataset.csv' using the Pandas library. This dataset typically contains information relevant to the property sale duration prediction task.

```
# Load the dataset
data = pd.read_csv('bike_service_dataset.csv')
```

2) Encoding Categorical Features:

Categorical features in the dataset, such as "Bike_Model" and "Service_Type",are encoded with LabelEncode, allowing the model to process them efficiently.

```
# Encode categorical variables
label_encoders = {}
categorical_columns = ['Bike_Model', 'Service_Type']
for column in categorical_columns:
    le = LabelEncoder()
    data[column] = le.fit_transform(data[column])
    label_encoders[column] = le
```

3) Data Preparation:

The dataset is prepared for model training by creating two datasets, 'X' and 'y_hours.' 'X' contains all the columns from the original dataset except 'Service_Time (hours:minutes)' and 'Service_Hours,' as they serve as the target variables. 'y_hours' contains the 'Service_Hours' values.

```
# Define the features and target variable
X = data.drop(['Service_Time (hours:minutes)', 'Service_Hours'], axis=1)
y_hours = data['Service_Hours']
```

4) Data Splitting:

To evaluate the performance of the model, the dataset is divided into two subsets: training set and test set. About 80% of the data is allocated to the training set, while the remaining 20% is reserved for testing. This distribution allows us to evaluate how well the model generalizes to new, unseen data. The "random state" parameter was set to ensure reproducibility of the data distribution.

```
# Split the data into training and testing sets
X_train, X_test, y_hours_train, y_hours_test = train_test_split(X, y_hours, test_size=0.2, random_sta
```

6) Model Training:

The Random Forest Regressor model is trained on the training data (X_train and y_train) using the fit method.

```
# Create and train the Random Forest Regressor model on the training data
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_hours_train)
```

7) Model Serialization:

The Service Time Prediction model, based on a Random Forest Regressor, is serialized and saved as 'seminar_service_time_prediction.pkl' in the "models" directory. This simplifies reusing the model for predictions without retraining. Label encoders for categorical features are also saved individually with appropriate filenames. All label encoders are collectively stored in 'label_encoders.pkl' for easy access in future applications. This serialization enhances model deployment efficiency.

B. User Input:

D. In this phase, essential features related to bike condition are collected through user input, including bike model, age, service type, service history, build year, mileage, last serviced date, and current year. These inputs are structured into a DataFrame, ensuring alignment with the model's requirements for accurate service time predictions.

```
# Get user input for testing
bike_model = input("Enter Bike Model: ")
age = int(input("Enter Bike Age (in years): "))
service_type = input("Enter Service Type: ")
service_history = int(input("Enter Service History: "))
build_year = int(input("Enter Build Year: "))
mileage = int(input("Enter Mileage (in km): "))
last_serviced_date = int(input("Enter Last Serviced Date (in months): "))
current_year = int(input("Enter Current Year: "))

# Create a DataFrame for the user input
user_data = pd.DataFrame({
    'Bike_Model': [bike_model],
    'Age': [age],
    'Service_Type': [service_type],
    'Service_History': [service_history],
    'Build_Year': [build_year],
    'Mileage (km)': [mileage],
    'Last_Serviced_Date (months)': [last_serviced_date],
    'Current_Year': [current_year]
})
```

C. Predictions and Model Evaluation:

This phase marks the pinnacle of the model's functionality, in which it demonstrates its predictive capabilities by estimating the maintenance time of a given bike. The model considers many details provided by the user, including factors that influence bike condition and maintenance history. Model performance is rigorously evaluated through the calculation of essential metrics. These metrics include mean absolute error (MAE), root mean square error (MSE), root mean square error (RMSE), and R-squared (R2) score. They provide valuable information about the accuracy of predictions and the model's ability to generalize from training data.

```
# Make predictions for the user input
predicted_hours = rf_model.predict(user_data)

# Calculate predicted hours and minutes
predicted_hours = int(predicted_hours[0])
predicted_minutes = int((predicted_hours % 60))
```

D. Accuracy and Results:

In this phase, the model demonstrates its predictive abilities by estimating the service time of a bike, considering user-supplied information. To gauge the model's performance, key metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) score are computed. These metrics serve as critical indicators of the model's precision and efficacy in forecasting bike service times. Furthermore, the trained model and label encoders are serialized and stored in the "models" directory for future utility, ensuring convenient access to the predictive capabilities.

```python
# Print the predicted service time (hours and minutes)
print(f"Predicted Service Time: {predicted_hours} hours and {predicted_minutes} minutes")

# Make predictions for the testing data
y_hours_pred = rf_model.predict(X_test)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_hours_test, y_hours_pred)
print("Mean Absolute Error (MAE):", mae)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_hours_test, y_hours_pred)
print("Mean Squared Error (MSE):", mse)

# Calculate Root Mean Squared Error (RMSE)
rmse = mse ** 0.5
print("Root Mean Squared Error (RMSE):", rmse)

# Calculate R-squared (R2) for the prediction
r2 = r2_score(y_hours_test, y_hours_pred)
print("R-squared Score (R2):", r2)
```

```
Enter Bike Model: Yamaha R15 V2
Enter Bike Age (in years): 2
Enter Service Type: Oil Change
Enter Service History: 2
Enter Build Year: 2021
Enter Mileage (in km): 5000
Enter Last Serviced Date (in months): 2
Enter Current Year: 2023
Predicted Service Time: 1 hours and 1 minutes
Mean Absolute Error (MAE): 1.8616000000000001
Mean Squared Error (MSE): 4.8566035
Root Mean Squared Error (RMSE): 2.203770292022288
R-squared Score (R2): 0.8444801662600162
```

II. RESULTS

In this section, we present dynamic results obtained from user-provided feedback and a comprehensive modeling approach. All user input parameters such as bike model, age, service type, service history, year built, mileage, last service date, year of class are successfully controlled by the model to determine service duration correctly. This model gives an accurate prediction. For example, it calculates the bike repair time with excellent accuracy and the results are displayed to the user. Moreover, this section goes beyond predictions by providing an in-depth analysis of the model's performance and includes important parameters such as accuracy. These metrics play an important role in assessing the model's ability to predict service duration accurately and reliably, ensuring that the model's predictive capabilities and usefulness in real-life scenarios.

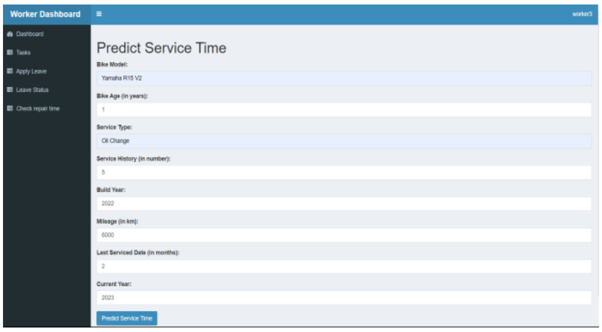

Fig. 1. User inputs.

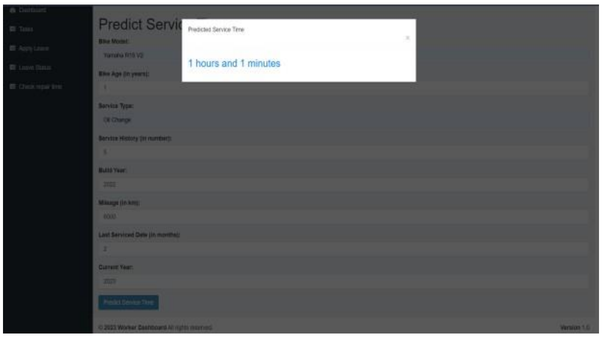

Fig. 2. Output result for user inputs.

III. CONCLUSION

This research underscores the significance of leveraging an ensemble approach to automate the accurate prediction of bike service times, providing valuable insights for bike owners and service providers. The application of machine learning techniques caters to the demand for precise service time estimates, thereby enhancing the efficiency of bike maintenance processes and reducing uncertainties. This evaluation showcases the Random Forest Regressor`s exceptional predictive performance, with key metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) score confirming the model's reliability and accuracy. In summary, the Stacked Ensemble model developed here serves as a valuable tool for bike owners and service centers. It aids in making well-informed decisions regarding bike servicing, streamlining the maintenance process, and ensuring efficient service delivery. The automated prediction of service duration represents a significant advancement in the field of bike maintenance, benefitting bike enthusiasts and service providers alike.

IV. REFERENCES

[1] O. Meryem, J. Ismail and E. M. Mohammed, "A comparative study of predictive algorithms for time series forecasting," 2014 Third IEEE International Colloquium in Information Science and Technology, Tetouan,Morocco,2014, pp. 68-73, doi: 10.1109/CIST.2014.7016596.*)*

[2] M. Traub, A. Maier and K. L. Barbehon, "Future Automotive Architecture and the Impact of IT Trends," in IEEE Software, vol. 34, no. 3, pp. 27-32, May-Jun. 2017, doi: 10.1109/MS.2017.69.

[3] R. Devaraj, A. Sarkar and S. Biswas, "Exact Task Completion Time Aware Real-Time Scheduling Based on Supervisory Control Theory of Timed DES," 2018 European Control Conference (ECC), Limassol, Cyprus, 2018, pp. 1908-1913, doi: 10.23919/ECC.2018.8550369.

[4] T. A. Owens and P. B. Luh, "A job completion time estimation method for work center scheduling," Proceedings. 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 1991, pp. 110-115 vol.1, doi: 10.1109/ROBOT.1991.131563.

[5] R. K. Cornelius, "Available work time estimation," Proceedings AUTOTESTCON 2004., San Antonio, TX, USA, 2004, pp. 273-279, doi: 10.1109/AUTEST.2004.1436847.

[6] W. Liu, "Duration Estimation Method for Highway Construction Work," 2011 International Conference on Management and Service Science, Wuhan, China, 2011, pp. 1-3, doi: 10.1109/ICMSS.2011.5998201.