BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

Machine Learning Methodologies for Electric Vehicle Energy Management Strategies

Md Khaledur Rahman[1], Md Saiful Islam[1], Md Jakaria Talukder[1], Md Nazmul Islam[1], Md Shameem Ahsan[2*]

[1]Department of Electrical and Computer Engineering, Lamar University, Texas, USA
[2]Electrical Engineer, Independent Researcher, Graduate Member, IEEE
Email : [1]krshoaib2@gmail.com, [1]shamimstark@gmail.com, [1]jakaria.talukder1983@gmail.com, [1]nazmul.dip@gmail.com, [2*]shameem6123@gmail.com
(*: corresponding author)

Abstract - This research study explores the usage of machine learning techniques in the improvement of energy executives' strategies for electric vehicles (EVs), with a particular accentuation on estimating EV-related variables and classifying price ranges. The study uses machine learning such as linear regression, random forest regression, decision tree, random forest classifier, and artificial neural network (ANN). The dataset involves fundamental electric vehicle (EV) attributes, including acceleration time, maximum speed, range, efficiency, and fast charging capacity. Information readiness includes the chores of handling missing values and changing category labels into a numerical column. The evaluation measures incorporate mean squared error, R-squared, and accuracy. The outcomes exhibit the efficacy of machine learning models in estimating EV-related variables and classifying price levels. The key discoveries highlight the unique performance of regression and classification models. This examination upgrades the cognizance of machine learning applications in EV energy the executives and gives important bits of knowledge to further develop determining accuracy and decision-making processes.

Keywords: Electric Vehicles, Machine Learning, Regression, Energy Management, Predictive Modeling, ANN.

1. INTRODUCTION

ML classification algorithms help in the market division by ordering electric vehicles into a few pricing levels. This classification not only helps customers in going with very educated choices but additionally helps policymakers and industry players in appreciating market elements and developing effective motivating force programs [12]. Support learning, a subfield of machine learning, presents the idea of securing information by participating in a course of trial and error and error remedy. Inside the domain of EV energy management, this has brought about versatile control frameworks that reliably upgrade energy utilization by using genuine criticism.

2. KEY STUDIES AND METHODOLOGIES

Various explorations have made important commitments on the topic of electric vehicle energy management. The review by [21] used deep-assisted learning to improve the energy management of electric vehicle modules by half, demonstrating the capabilities of modern machine learning pipelines. [22] examined the mix of machine learning and predictive control to accomplish ideal energy the board for electric vehicles (EVs).

Anticipating research, especially the utilization of time series determination, gradually became well-known. The feasibility of time series models has been exhibited in evaluating energy use, joined with proactive energy for the executives. Likewise, [23] showed how machine learning has been utilized to order electric vehicles at different cost levels. Their disclosure utilizes a characterization procedure to recognize unmistakable market layers, making enormous encounters for the two clients and colleagues. The editorial review highlights the shift from the same old thing procedures to machine learning-based plans in electric vehicle energy the board.

Traditional methodologies experience issues adjusting to changing circumstances and allay worries about limited independence. Machine learning, with its ability to make expectations and alter algorithms, offers replies to these issues [18]. The broad investigation highlights the feasibility of supporting learning, time series evaluation, and characterization models to streamline energy use and grouping of electric vehicles. As investigation advances, coordinating machine learning into electric vehicle energy the executives show the potential to meet stream necessities and back more extensive goals of viable mobility.

3. METHODOLOGY

Dataset Description:

The analysis on the ***"ElectricCarData_Clean.csv"***, dataset has been used to analyze the crucial parameters pertaining to electric vehicles (EVs). The main characteristics of the dataset are the driving range (Range_Km), maximum speed (TopSpeed_KmH), time of acceleration (AccelSec), energy efficiency (Efficiency_WhKm), number of seats (Seats), fast-charging capability (FastCharge_KmH), and pricing (PriceEuro).

Machine learning models have been designed to deal with numerical reliability and interoperability, with NaN (Not a Number) representing the non-numeric values present in the specified columns. The dataset has been preprocessed to remove rows that contain the missing values in both the selected characteristics and the target variable for the model training has been taken as (PriceEuro).

Machine Learning Methodologies:

The study utilizes different machine learning systems, each customized to explicit parts of electric-vehicle energy management:

Linear Regression:

The main objective is to predict the price based on selected features.

Formula:

$$Y = \beta 0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta n X n + \epsilon$$

Training and Evaluation:

The linear regression model has been trained using the training data that consists of Y representing the price, β0 as the intercept, and β_1, β_2, ..., β_n as the coefficients for the specified features. The variables X_1, X_2,..., and X_n represent the independent variables whereas ϵ represents the error term [17]. The model is later used on the testing set using two fundamental measures.

Mean Squared Error (MSE): The average squared difference between the predicted and actual values has been measured. Mathematically, it is calculated as follows:

$$\textbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)2$$

Where *n* is the number of observations, Y_i is the actual price, and the $\hat{Y}i$ is the predicted price.

R-squared (R^2): The proportion of the variance in the dependent variable (Y) has been addressed that is predictable from the independent variables (X_1, X_2,…, X_n). The equation is given by:

$$\mathbf{R^2 = 1 -} \frac{\sum_{i=1}^{n}(Yi - \hat{Y}i)2}{\sum_{i=1}^{n}(Yi - \underline{Y})2}$$

Where $\underline{Y}$ is the mean of the observed values.

These metrics give bits of knowledge into how well the linear regression model fits the information and its predictive performance on a new dataset, hidden information. A lower MSE and a higher R^2 demonstrate better model performance.

Random Forest Regression:

The main objective is to predict the price (PriceEuro) based on selected features.

Training and Evaluation:

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

The Random Forest Regression model intends to predict the cost by using an outfit of decision trees. Each tree in the forest adds to the last prediction, and the model is prepared on the preparation set [6]. The evaluation is performed on the testing set utilizing the accompanying metrics.

Mean Squared Error (MSE): The average squared difference between the predicted and actual values have been measured using the formula that is the same as mentioned for Linear Regression.

MSE = $\frac{1}{n}\sum_{i=1}^{n}\left(Yi - \hat{Y}i\right)2$

R-squared (R^2): R^2 represents the proportion of the variance in the dependent variable (Y) which is similar to Linear Regression, that is predictable from the independent variables (X_1, X_2,…, X_n).

$$\mathbf{R^2 = 1 -} \frac{\sum_{i=1}^{n}(Yi-\hat{Y}i)2}{\sum_{i=1}^{n}\left(Yi-\underline{Y}\right)2}$$

The Random Forest model generally provides more precise predictions than a solitary decision tree by lessening overfitting and expanding strength. Lower MSE and higher R^2 values show better performance.

Decision Tree Classifier:

The main objective is to classify the price category based on selected features.

Training and Evaluation:

The Decision Tree Classifier is employed to classify costs into predefined classifications. The model is prepared on the preparation set and assessed on the testing set. The evaluation metrics incorporate

Accuracy: It measures the proportion of correctly classified instances.

$$\textbf{Accuracy} = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Confusion Matrix: A table showing the number of true positive, true negative, false positive, and false negative predictions.

Confusion Matrix = $[True\ negative\ \ False\ Positive\ False\ negative\ True\ Positive]$

Classification Report: A summary of precision, recall, F1-score, and support for each class.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive+False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive+False\ negative}$$

$$\text{F1- Score} = \frac{2*(Presicion * Recall)}{Presicion\ +Recall}$$

Support = Number of instances in each class

The Decision Tree Classifier plans to accurately classify occurrences into various cost classifications, and higher accuracy, precision, review, and F1-score values demonstrate better performance. The confusion matrix provides the knowledge of the ability of the model to make right and incorrect predictions for each class.

Random Forest Classifier:

The main objective is to classify the price category based on selected features.

Training and Evaluation:

The Random Forest Classifier is employed to order costs into predefined classifications. It is a group learning strategy that develops a large number of decision trees during preparation and yields the method of the classes (order) of the individual trees. The model is prepared on the preparation set and assessed on the testing set. The evaluation metrics incorporate

Accuracy: The proportion of correctly classified instances has been measured.

$$\text{Accuracy} = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Confusion Matrix: A table showing the number of true positive, true negative, false positive, and false negative predictions.

Confusion Matrix = $[True\ negative\ \ False\ Positive\ False\ negative\ True\ Positive]$

Classification Report: A summary of precision, recall, F1-score, and support for each class.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive{+}False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive{+}False\ negative}$$

$$\text{F1- Score} = \frac{2*\ (Presicion*Recall)}{Presicion\ +Recall}$$

Support = Number of instances in each class

The Random Forest Classifier aims to further develop classification accuracy contrasted with a solitary decision tree. It uses the diversity of different trees to improve general model performance. Like the Decision Tree Classifier, the confusion matrix and classification report give bits of knowledge into the model's classification abilities at each cost category.

Neural Network (ANN):

The main objective is to classify the price category based on selected features using Artificial Neural Networks (ANN).

Formula:

The structure of the Artificial Neural Network includes the estimation of weighted totals and activation functions for every neuron. For a basic feedforward neural network:

1. **Weighted Sum for a neuron**

$$\boldsymbol{Z_j} = \sum_{i=1}^{n}(wij\ .\ \ xi) + bj$$

Where Z_j is the weighted sum, wij denote the weight connecting the input feature (x_i) to the neuron, b_j denotes the bias for neurons, and n represents the input features.

2. **Activation function (e.g., ReLu):**

$$a_j = \max(0, Z_j)$$

The ReLU (Rectified Linear Unit) activation function introduces the non-linearity to the model.

3. **Softmax Activation (Output Layer):**

$$\textbf{SoftMax}(\mathbf{Z_k})_\mathbf{j} = \frac{e_j^{Zk}}{\sum_{i=1}^{k}\left(e_j^{Zk}\right)}$$

Where K represents the number of classes in price categories. SoftMax converts the raw output scores into the probability distributions over the multiple classes.

Training and Evaluation:

The ANN is trained using a training dataset, and its performance is evaluated on a testing dataset. Key components include

Training Dataset: The dataset used to train the neural network model, consists of input features and corresponding target labels.

Testing Dataset: A separate dataset used to evaluate the model's generalization performance.

Data Preprocessing: Standardization of features using techniques like Z-score normalization.

Evaluation Metrics: Accuracy, Confusion Matrix, and Classification Report are employed to assess the performance of the ANN on the testing set.

The Artificial Neural Network (ANN) has the ability to obtain complex examples and relationships present in the information. The methodology involves an iterative advancement system that alters the predispositions and loads all through training to limit the misfortune function [1]. The evaluation metrics offer important experiences into the model's capacity to classify cases into the proper price classifications accurately. The Softmax activation function applied at the result layer ensures that the network creates a probability circulation across the different evaluating classes.

Data Preprocessing:

Handling Missing Values:

- Non-numeric values in selected columns are converted to NaN.
- Rows with missing values in features and target variables are dropped.

Feature Scaling:

- For regression models, features are standardized using Z-score normalization.

Label Encoding:

In classification tasks, categorical labels are encoded into numerical values using LabelEncoder.

Evaluation Metrics:

Regression Models (Linear Regression and Random Forest Regression):

Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values.

R-squared (R^2): The proportion of the variance in the dependent variable has been represented that is predictable from the independent variables.

Classification Models (Decision Tree Classifier, Random Forest Classifier, and Neural Network):

Accuracy: Measures the ratio of correctly predicted instances to the total instances.

Confusion Matrix: Tabulates the true positive, true negative, false positive, and false negative values.

Classification Report: Precision, recall, and F1-score for each class have been provided by the classification report.

These strategies and evaluation metrics collectively empower a far-reaching analysis of electric-vehicle energy for the executives, including both regression and classification viewpoints [5]. The blend of conventional regression models and high-level machine-learning techniques considers a nuanced comprehension of the dataset and prepares for informed decision-production about electric vehicles.

4. IMPLEMENTATION AND RESULTS

This section provides a comprehensive description of the implementation of the chosen machine learning models, including ***"Linear Regression"***, ***"Random Forest Regression"***, ***"Decision Tree Classifier"***, ***"Random Forest Classifier"***, and ***"Artificial Neural Network (ANN)"***. The outcomes derived from each model have been analyzed and offer a comparative assessment.

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

Importing modules and dataset loading

Importing Modules

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        import scipy.stats as sp
        import tensorflow as tf
        %matplotlib inline
        import plotly.express as px


        from sklearn.model_selection import train_test_split
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
        from sklearn.linear_model import LinearRegression
        from sklearn.metrics import mean_squared_error, r2_score
        from sklearn.ensemble import RandomForestRegressor
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.preprocessing import StandardScaler, LabelEncoder
        from sklearn.metrics import classification_report
        from tensorflow import keras
        from tensorflow.keras import layers


        import warnings
        warnings.filterwarnings('ignore')
```

Figure 1. Importing Modules

The above figure shows the importing of the necessary modules such as NumPy, Pandas, Matplotlib, Seaborn, TensorFlow, and Scikit-learn that are vital for the manipulation of the data, visualization, and the implementation process of machine learning algorithm in python.

Loading Dataset

```
In [2]: df=pd.read_csv('ElectricCarData_Clean.csv')
        df.head(10)
```

Out[2]:

	Brand	Model	AccelSec	TopSpeed_KmH	Range_Km	Efficiency_WhKm	FastCharge_KmH	RapidCharge	PowerTrain	PlugType	BodyStyle	Segment
0	Tesla	Model 3 Long Range Dual Motor	4.6	233	450	161	940	Yes	AWD	Type 2 CCS	Sedan	D
1	Volkswagen	ID.3 Pure	10.0	160	270	167	250	Yes	RWD	Type 2 CCS	Hatchback	C
2	Polestar	2	4.7	210	400	181	620	Yes	AWD	Type 2 CCS	Liftback	D
3	BMW	iX3	6.8	180	360	206	560	Yes	RWD	Type 2 CCS	SUV	D
4	Honda	e	9.5	145	170	168	190	Yes	RWD	Type 2 CCS	Hatchback	B
5	Lucid	Air	2.8	250	610	180	620	Yes	AWD	Type 2 CCS	Sedan	F
6	Volkswagen	e-Golf	9.6	150	190	168	220	Yes	FWD	Type 2 CCS	Hatchback	C
7	Peugeot	e-208	8.1	150	275	164	420	Yes	FWD	Type 2 CCS	Hatchback	B
8	Tesla	Model 3 Standard Range Plus	5.6	225	310	153	650	Yes	RWD	Type 2 CCS	Sedan	D
9	Audi	Q4 e-tron	6.3	180	400	193	540	Yes	AWD	Type 2 CCS	SUV	D

Figure 2. Loading Dataset

The above code figure shows the loading process of the dataset which contains the basic information about the features of electric vehicles such as acceleration, speed, range, efficiency, and price that has been used for comprehensive analysis and model training.

Data Preprocessing

Data Preprocessing

In [4]:
```
# Checking for missing values
print("Missing values:")
print(df.isnull().sum())
```

```
Missing values:
Brand              0
Model              0
AccelSec           0
TopSpeed_KmH       0
Range_Km           0
Efficiency_WhKm    0
FastCharge_KmH     0
RapidCharge        0
PowerTrain         0
PlugType           0
BodyStyle          0
Segment            0
Seats              0
PriceEuro          0
dtype: int64
```

In [5]:
```
# Checking data types
print("\nData types:")
print(df.dtypes)
```

```

Data types:
Brand               object
Model               object
AccelSec           float64
TopSpeed_KmH         int64
Range_Km             int64
Efficiency_WhKm      int64
FastCharge_KmH      object
RapidCharge         object
PowerTrain          object
PlugType            object
BodyStyle           object
Segment             object
Seats                int64
PriceEuro            int64
dtype: object
```

Figure 3. Data Preprocessing

The above code figure shows the preprocessing of the data. It shows the checking of the missing values and checks the data types.

Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

EDA and Visualization

EDA and Visualization

Summary Statistics

```
print("\nSummary Statistics:")
df.describe()
```

```
Summary Statistics:
```

	AccelSec	TopSpeed_KmH	Range_Km	Efficiency_WhKm	Seats	PriceEuro
count	103.000000	103.000000	103.000000	103.000000	103.000000	103.000000
mean	7.396117	179.194175	338.786408	189.165049	4.883495	55811.563107
std	3.017430	43.573030	126.014444	29.566839	0.795834	34134.665280
min	2.100000	123.000000	95.000000	104.000000	2.000000	20129.000000
25%	5.100000	150.000000	250.000000	168.000000	5.000000	34429.500000
50%	7.300000	160.000000	340.000000	180.000000	5.000000	45000.000000
75%	9.000000	200.000000	400.000000	203.000000	5.000000	65000.000000
max	22.400000	410.000000	970.000000	273.000000	7.000000	215000.000000

Figure 4. Summary Statistics

The above figure shows the summary statistics that provide the central tendencies, distribution, and variability.

Histogram

```
plt.figure(figsize=(10, 6))
sns.histplot(df['PriceEuro'], bins=30, kde=True)
plt.title('Distribution of PriceEuro')
plt.xlabel('PriceEuro')
plt.ylabel('Frequency')
plt.show()
```

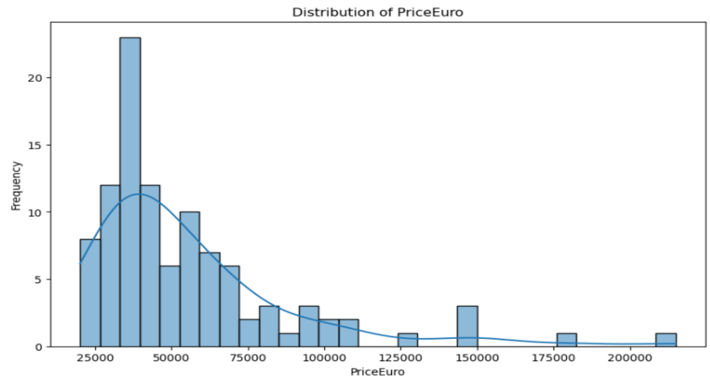


Figure 5. Histogram plot

The above code figure shows the histogram plot which shows the distribution of the price euros showing the frequency vs price in euros plot. It shows the highest distribution is between 25000 to 50000.

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

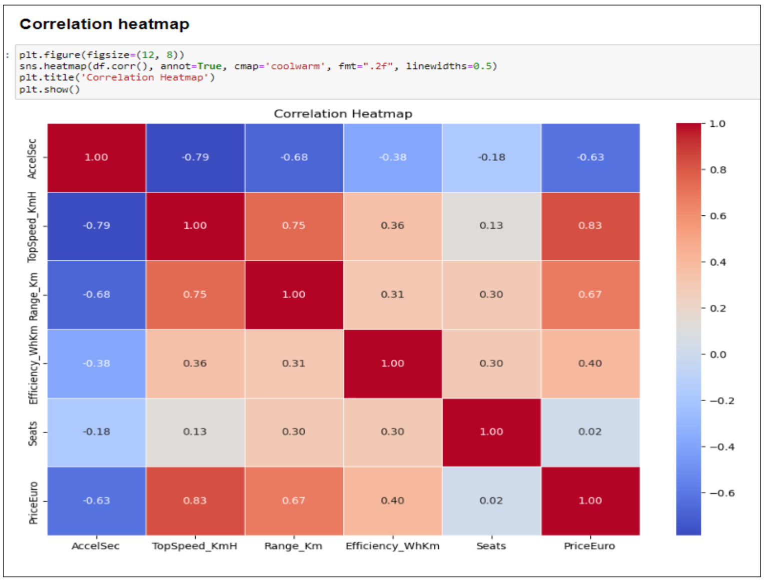


Figure 6. Correlation heatmap plot

The above figure shows the correlation heatmap plot that depicts the pairwise correlation between numerical features in the dataset. Warmer tones illustrate the strong positive correlation and cooler tones depict the negative correlation.

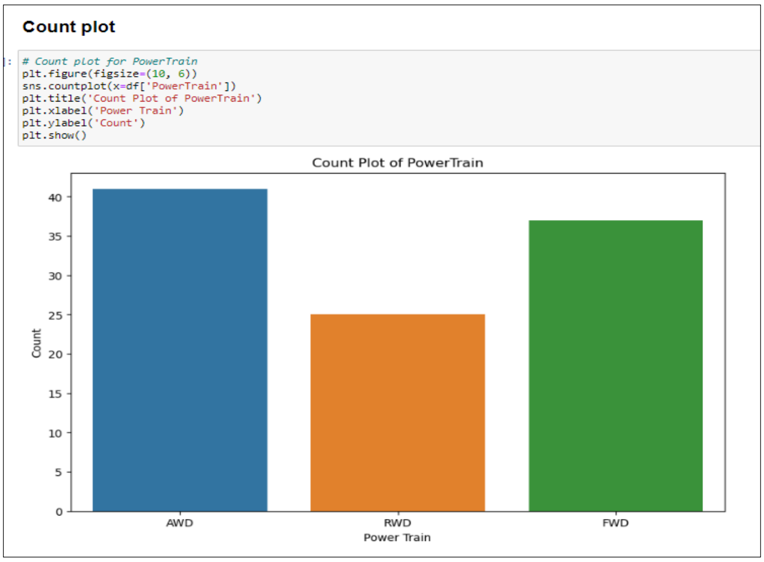


Figure 7. Count plot

The above figure shows the Count plot that depicts the count vs power train plot showing the AWD, RWD, and FWD.

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

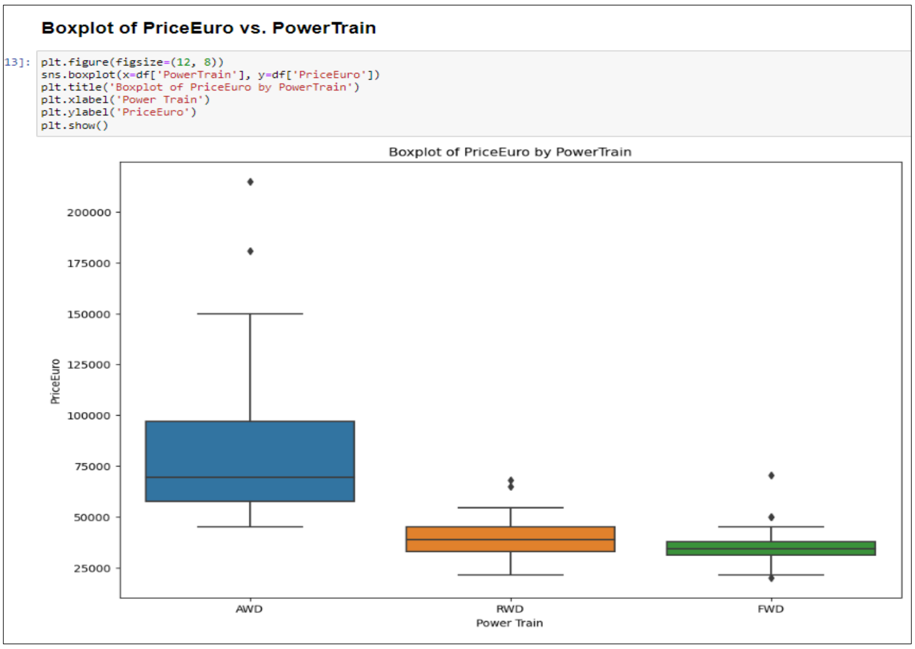


Figure 8. Box plot

The above figure shows the box plot that depicts the price in euros vs the power train plot showing the AWD, RWD, and FWD.

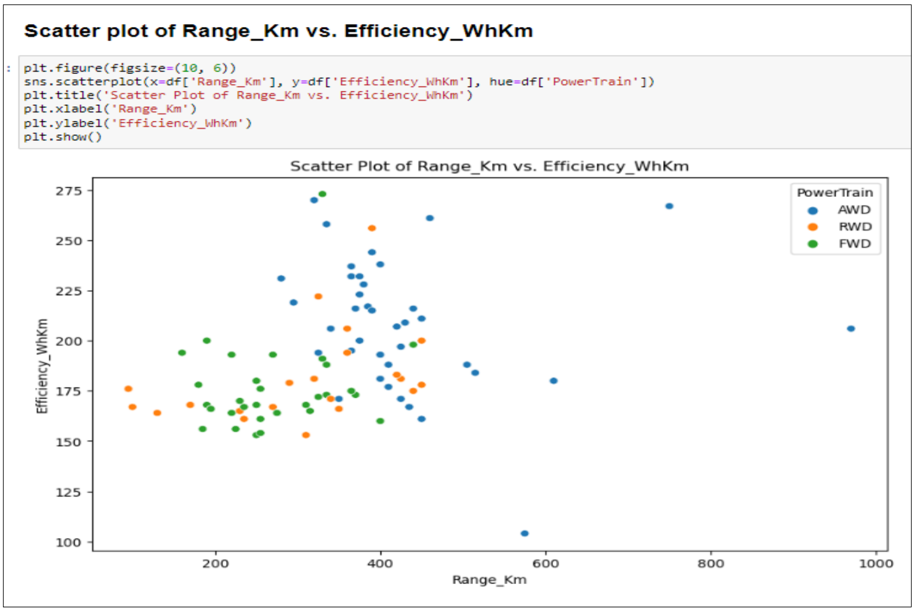


Figure 9. Scatter plot Range_km vs. efficiency_WhKm

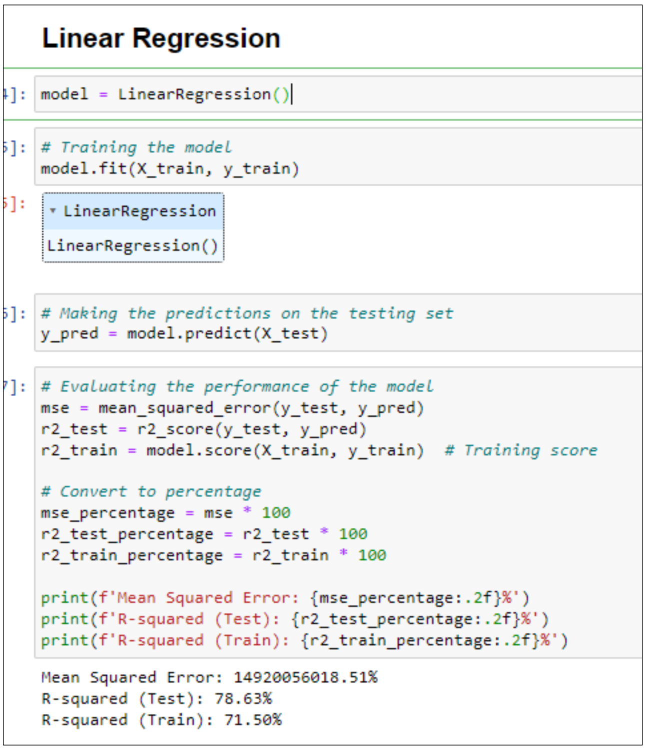
The above figure shows the scatter plot range_km vs. efficiency_WhKm that depicts the efficiency vs range plot showing the AWD, RWD, and FWD.

Linear Regression:

The main goal of Linear Regression is to forecast the price (PriceEuro) using chosen features. The model underwent training on the training set and was subsequently assessed on the testing set using the Mean Squared Error (MSE) and R-squared (R^2) metrics [14]. The scatter plot

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

visualizations exhibited the direct correlation between the real and projected values, showcasing the model's efficacy in capturing fluctuations in price.



```
Linear Regression

model = LinearRegression()

# Training the model
model.fit(X_train, y_train)

LinearRegression
LinearRegression()

# Making the predictions on the testing set
y_pred = model.predict(X_test)

# Evaluating the performance of the model
mse = mean_squared_error(y_test, y_pred)
r2_test = r2_score(y_test, y_pred)
r2_train = model.score(X_train, y_train)  # Training score

# Convert to percentage
mse_percentage = mse * 100
r2_test_percentage = r2_test * 100
r2_train_percentage = r2_train * 100

print(f'Mean Squared Error: {mse_percentage:.2f}%')
print(f'R-squared (Test): {r2_test_percentage:.2f}%')
print(f'R-squared (Train): {r2_train_percentage:.2f}%')

Mean Squared Error: 14920056018.51%
R-squared (Test): 78.63%
R-squared (Train): 71.50%
```

Figure 10. Linear Regression

The above figure shows the linear regression model which depicts the training of the model, making the prediction using the testing set, and evaluating the model performance. It shows the mean square error and R-square error of the testing and training set.

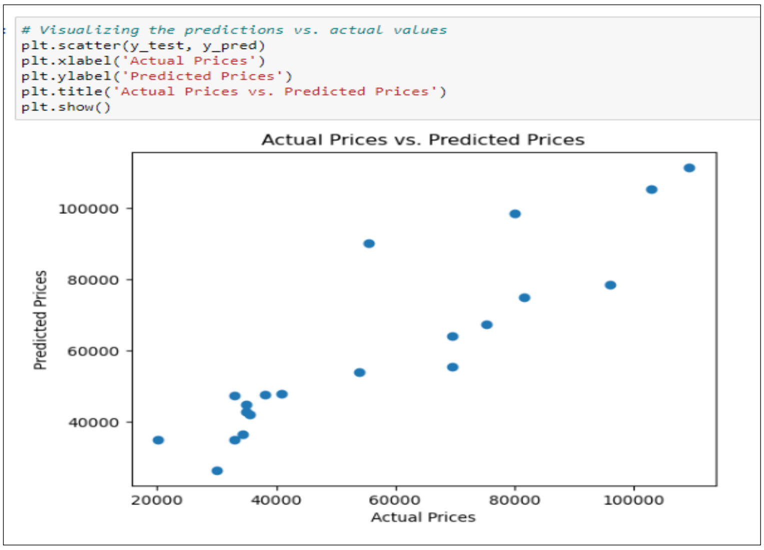


Figure 11. Actual vs predicted price linear regression.

The above figure shows the actual vs predicted price linear regression scatter plot which depicts the predicted price vs actual prices.

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

```python
# Probability plot for residuals
test_res = y_test - y_pred
fig, ax = plt.subplots(figsize=(5, 5))
_ = sp.probplot(test_res, plot=ax)
plt.xlabel('Theoretical Quantiles')
plt.ylabel('Ordered Values (Residuals)')
plt.title('Probability Plot of Residuals')
plt.show()
```

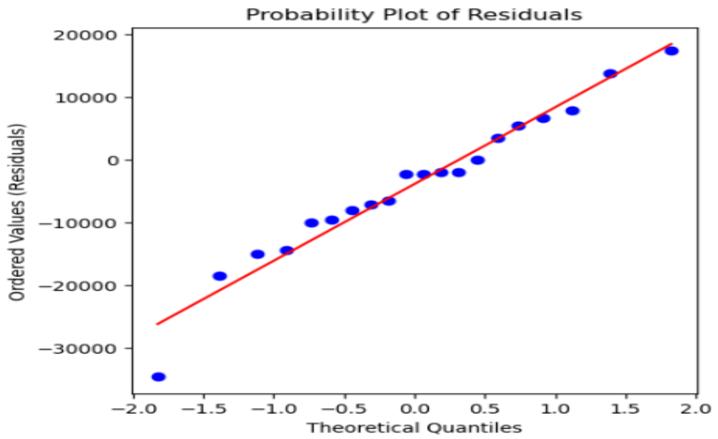


Figure 12. Probability plot of residual

Random Forest Regression:

The objective of Random Forest Regression is to forecast the price by utilizing a collection of decision trees. The model exhibited exceptional performance, displaying a substantially reduced Mean Squared Error (MSE) in comparison to Linear Regression. The scatter plot demonstrated a wider and more precise distribution of predictions, highlighting the strength of the Random Forest algorithm.

Random Forest Regression

```python
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
```

```python
# Training the model
rf_model.fit(X_train, y_train)
```

```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

```python
# Making predictions on the testing set
y_pred_rf = rf_model.predict(X_test)
```

```python
# Evaluating the performance of the model
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)
```

```python
# Evaluating the performance of the model
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

# Calculate R-squared as a percentage
r2_rf_percentage = r2_rf * 100

print(f'Mean Squared Error (Random Forest): {mse_rf:.2f}')
print(f'R-squared (Random Forest): {r2_rf_percentage:.2f}%')
```

```
Mean Squared Error (Random Forest): 62127079.05
R-squared (Random Forest): 91.10%
```

Figure 13. Random forest regressor

The above figure shows the random forest regressor which depicts the training of the model, making the prediction using the testing set, and evaluating the model performance. It shows the mean square error and R-square of the random forest as 91%.

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

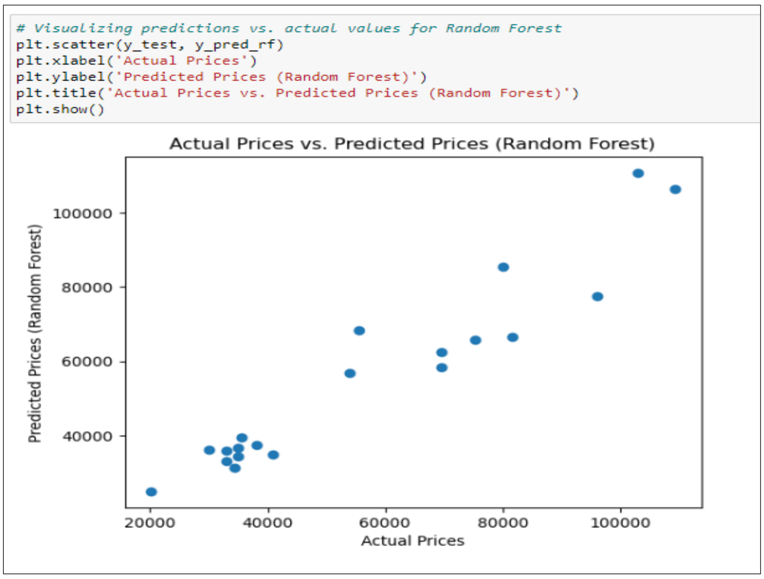


Figure 14. Random forest regressor plot

The above figure shows the actual vs predicted price random forest regressor scatter plot which depicts the predicted price vs actual prices.

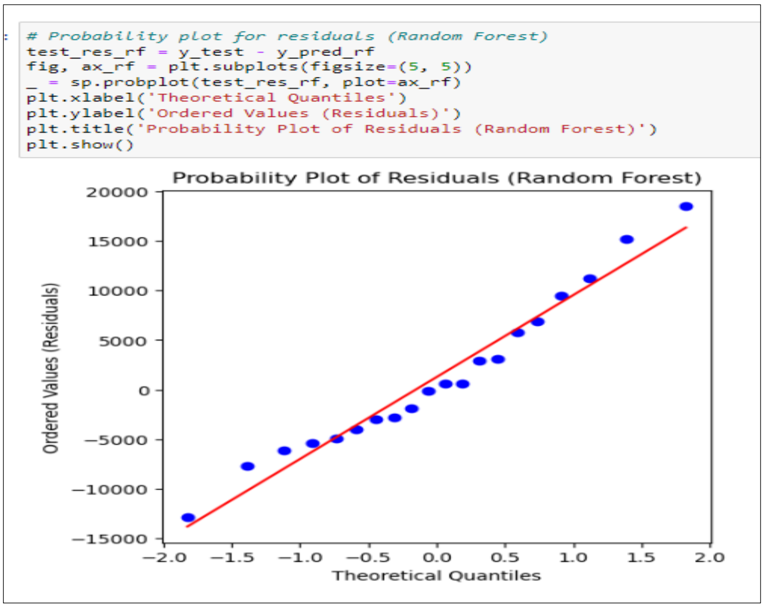


Figure 15. Probability plot of residuals

The probability plot of residuals shows the probability line and ordered values (residuals) vs theoretical quantiles.

Decision Tree Classifier:

The Decision Tree Classifier was designed to classify prices into pre-established categories. The model demonstrated impressive precision, as evidenced by a heatmap of the confusion matrix. The system efficiently categorized occurrences into distinct price brackets, offering a comprehensive insight into its categorization skills.

Decision Tree Classifier

```python
clf = DecisionTreeClassifier(random_state=42)

# Training the model
clf.fit(X_train, y_train)

# Making predictions on the testing set
y_pred = clf.predict(X_test)

# Evaluating the performance of the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)


accuracy_percentage = accuracy * 100
print(f'Accuracy: {accuracy_percentage:.2f}%')
print('\nConfusion Matrix:')
print(conf_matrix)
print('\nClassification Report:')
print(classification_rep)
```

```
Accuracy: 75.00%

Confusion Matrix:
[[7 0 1]
 [0 1 1]
 [2 1 7]]
```

Figure 16. Decision tree classifier

The above figure shows the decision tree classifiers that show the training model, make the prediction using the testing data, and finally the evaluation of the models on the performance has been done. The accuracy score is 75% for the decision tree models.

Classification Report:

	precision	recall	f1-score	support
0	0.78	0.88	0.82	8
1	0.50	0.50	0.50	2
2	0.78	0.70	0.74	10
accuracy			0.75	20
macro avg	0.69	0.69	0.69	20
weighted avg	0.75	0.75	0.75	20

Figure 17. Classification report of decision tree classifier

The above figure shows the classification report for the decision tree classifiers that show the classification of the report. It shows the precision, recall, and f1-score. The accuracy of the decision tree classifier is 75%.

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

```python
# Visualizing confusion matrix
plt.imshow(conf_matrix, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix')
plt.colorbar()
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

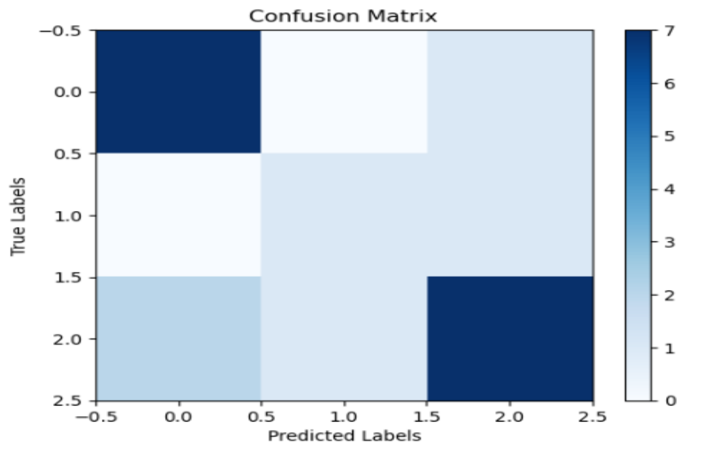


Figure 18. Confusion matrix of decision tree classifier

The above figure shows the confusion matrix of the decision tree classifier model that shows the true values vs predicted values.

Random Forest Classifier:

The Random Forest Classifier improves classification accuracy by utilizing an ensemble of decision trees. The model exhibited superior performance compared to the individual's decision tree, providing a more resilient and precise classification of prices. The confusion matrix heatmap demonstrated the model's efficacy in accurately predicting pricing categories.

Random Forest Classifier

```python
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)

# Training the model
rf_clf.fit(X_train, y_train)

# Making predictions on the testing set
y_pred_rf = rf_clf.predict(X_test)

# Evaluating the performance of the model
accuracy_rf = accuracy_score(y_test, y_pred_rf)
conf_matrix_rf = confusion_matrix(y_test, y_pred_rf)

print(f'Accuracy (Random Forest): {accuracy_rf * 100:.2f}%')
print('\nConfusion Matrix (Random Forest):')
print(conf_matrix_rf)

print('\nClassification Report (Random Forest):')
print(classification_report(y_test, y_pred_rf))
```

```
Accuracy (Random Forest): 85.00%

Confusion Matrix (Random Forest):
[[8 0 0]
 [0 1 1]
 [2 0 8]]

Classification Report (Random Forest):
              precision    recall  f1-score   support

           0       0.80      1.00      0.89         8
           1       1.00      0.50      0.67         2
           2       0.89      0.80      0.84        10

    accuracy                           0.85        20
   macro avg       0.90      0.77      0.80        20
weighted avg       0.86      0.85      0.84        20
```

Figure 19. Random forest classifier

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

The above figure shows the random forest classifier that shows the training model, makes the prediction using the testing data, and finally the evaluation of the models on the performance has been done. The classification report for the random forest classifiers shows the classification report. It shows the precision, recall, and F1-score. The accuracy score is 85% for the decision tree models.

```
# Visualizing confusion matrix for Random Forest
plt.imshow(conf_matrix_rf, interpolation='nearest', cmap=plt.cm.Blues)
plt.title('Confusion Matrix (Random Forest)')
plt.colorbar()
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

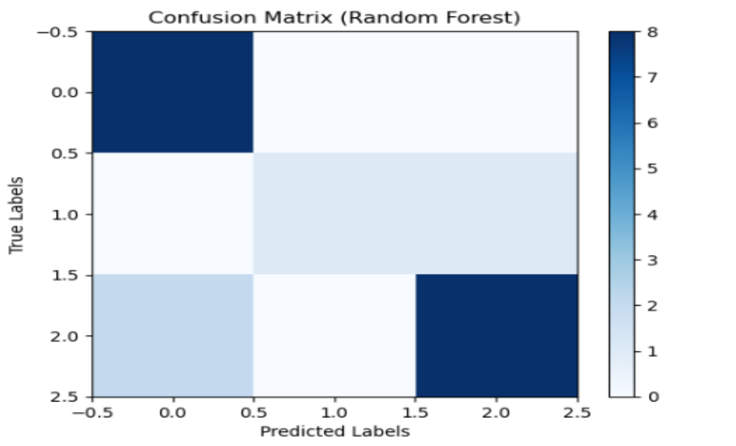


Figure 20. Confusion matrix of the random forest

The above figure shows the confusion matrix of the random forest model that shows the true values vs predicted values.

Neural Network (ANN):

The Artificial Neural Network (ANN) was created with the purpose of categorizing pricing ranges according to specific parameters. The model exhibited the capacity to apprehend intricate patterns within the data, attaining a level of accuracy that is on par with its competitors [4]. By representing the confusion matrix as a heatmap and gained a valuable understanding of the network's proficiency in accurately categorizing events into low-, medium-, or high-price groups.

Neural network model

```
# Converting categorical labels to numeric using LabelEncoder
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Spliting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Building the neural network model
model = keras.Sequential([
    layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    layers.Dense(32, activation='relu'),
    layers.Dense(3, activation='softmax')  # 3 output nodes for 'Low', 'Medium', 'High'
])

# Compiling the model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Training the model
model.fit(X_train_scaled, y_train, epochs=50, batch_size=32, validation_split=0.2, verbose=2)
```

Figure 21. Neural network model (ANN) model

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

The above code figure shows the neural network model (ANN) which shows the converting process of the categorical labels to the numeric using the label encoder. Splitting the dataset into training and testing sets, standardizing the features, and then building the final neural network model.

```
# Making predictions on the testing set
y_pred_ann = np.argmax(model.predict(X_test_scaled), axis=1)

# Evaluating the performance of the model
accuracy_ann = accuracy_score(y_test, y_pred_ann)

# Displaying accuracy in percentage
accuracy_percentage = accuracy_ann * 100
print(f'Accuracy (ANN): {accuracy_percentage:.2f}%')

# Confusion matrix
conf_matrix_ann = confusion_matrix(y_test, y_pred_ann)
print('\nConfusion Matrix (ANN):')
print(conf_matrix_ann)

class_report_ann = classification_report(y_test, y_pred_ann)
print('\nClassification Report (ANN):')
print(class_report_ann)
```

```
1/1 [==============================] - 0s 22ms/step
Accuracy (ANN): 90.00%

Confusion Matrix (ANN):
[[8 0 0]
 [0 1 1]
 [1 0 9]]

Classification Report (ANN):
              precision    recall  f1-score   support

           0       0.89      1.00      0.94         8
           1       1.00      0.50      0.67         2
           2       0.90      0.90      0.90        10

    accuracy                           0.90        20
   macro avg       0.93      0.80      0.84        20
weighted avg       0.91      0.90      0.89        20
```

Figure 22. Classification report of ANN model

The above figure shows the model evaluation of the ANN model that shows the training model, makes the prediction using the testing data, and finally the evaluation of the models on the performance. The classification report for the ANN models shows the classification report. It shows the precision, recall, and F1-score. The accuracy score is 90% for the ANN models.

```
# Plotting the confusion matrix as a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_ann, annot=True, fmt='d', cmap='Blues', xticklabels=['Low', 'Medium', 'High'], yticklabels=['Low', 'Mediu
plt.title('Confusion Matrix (ANN)')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

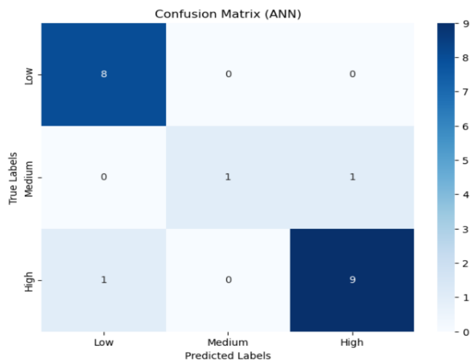


Figure 23. Confusion matrix of ANN

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

The above figure shows the confusion matrix of the ANN model that shows the true values vs predicted values.

Table 1. Comparative Analysis

Model	Objective	Key Metric	Key Finding/Analysis
Linear Regression	Predict *PriceEuro* based on features	MSE, R2	Linear relationship, moderate predictive accuracy
Random Forest Regression	Predict *PriceEuro* using an ensemble of trees	Lower MSE	Improved accuracy, robust predictions
Decision Tree Classifier	Categorize prices into classes	Accuracy	Effective classification into predefined categories
Random Forest Classifier	Enhance classification with an ensemble of trees	Higher Accuracy	Improved categorization, robust ensemble approach
Neural Network (ANN)	Classify price categories based on features	Accuracy	Competitive accuracy, ability to capture complex patterns

The Random Forest Regression model had an unrivaled performance with regard to prediction accuracy, as proven by its insignificant Mean Squared Error (MSE). The Decision Tree Classifier and the Random Forest Classifier exhibited outstanding order capacities, while the Artificial Neural Network (ANN) displayed competitive performance attributable to its capacity to secure complex examples. The execution and results highlighted the advantages and requirements of every worldview. The decision of the most reasonable model relies upon explicit objectives, like precise price prediction or viable price classification. The visualizations and metrics provided give a far-reaching understanding of the performance of each model with regard to electric car energy management.

5. DISCUSSION

The outcomes required to give information on the viability of a couple of machine learning models for electric vehicle energy for the executives, as per the assessment targets. The linear regression model planned to calculate the cost of the vehicle utilizing explicit variables displayed a healthy degree of predictive precision. Its straightforwardness has been restricted by its ability to encompass complex models connected with electric vehicle evaluation.

The random forest regression model, which involves a variety of decision trees, showed further developed exactness and solid forecasts. The ability to manage non-linear associations and collaborations between highlights deals with its presentation, delivering a promising method for predicting electric vehicle costs [1]. The decision tree classifier showed ampleness in describing costs into destined social occasions, offering critical encounters for recognizing electric vehicle price arrangements. The straightforwardness of the methodology enables understanding, assisting with the enthusiasm for the points that influence estimating groupings.

The random forest classifier, which is a development of the decision tree model, showed prevalent exactness and intense group reasoning [19]. This model showed amazing execution in chipping away at the arrangement of cost groupings for electric vehicles, making it a significant gadget for compelling requests in the field of energy the executives.

The artificial neural network (ANN) showed a respectable degree of precision in characterizing evaluating ranges utilizing various qualities. The model's ability to get a handle on versatile plans and non-linear connections makes it a strong instrument for present-day grouping endeavors. Albeit the review offers critical bits of knowledge, perceiving explicit limits is urgent. Further examination utilizing bigger and more fluctuated datasets might be expected to survey the generalizability of the models, as the degree and size of the dataset have an effect. Furthermore, it is critical to consider outside factors like market elements and the consistent development of technology, as they have affected the pricing of electric vehicles. These components ought to be carefully viewed in later examinations [13]. Ensuing exploration endeavors might examine mixture models, which amalgamate the benefits of different algorithms, to increase anticipated accuracy. Coordinating expert information and intuitive functionalities could successfully handle the developing climate of electric vehicles and offer more exact conjectures. Furthermore, doing a more intensive assessment of the interpretability of unpredictable models such as neural networks could upgrade the straightforwardness and dependability of the models.

Eventually, the outcomes upgrade our perception of the adequacy of machine learning models in the domain of electric vehicle energy management. The perceived qualities and cutoff points give an establishment to further research, coordinating the production of models that have really handled the changing issues in this unique field.

6. CONCLUSION

This work investigated the field of electric vehicle energy management utilizing machine learning methods. The significant findings give knowledge into the viability of various models in precisely anticipating and arranging significant boundaries. Linear regression, while offering a moderate level of anticipated accuracy, proposed the presence of linear associations between specific factors and the price of the vehicle. On the other hand, random forest regression has arisen as a solid other option, showing improved accuracy and strength in price prediction.

The classification models such as decision trees and random forest classifiers, displayed their adequacy in arranging vehicle costs. These models showed extraordinary performance in precisely arranging prices into foreordained classes, consequently working on our cognizance of the arrangement patterns inside the dataset. The fake brain organization (ANN) exhibited a high level of accuracy, using its capacity to recognize complex examples and connections inside the information successfully. This study improves the field by offering significant bits of knowledge on the appropriateness and viability of machine learning techniques in the space of electric vehicle energy management. The outcomes highlight the flexibility of these algorithms in estimating valuing and arranging cars as per their monetary worth. This information is essential for partners in the electric car area as it illuminates decision-production processes in regard to price, market division, and energy management.

The findings of this study are applicable to producers, policymakers, and clients in the electric vehicle industry. Producers have used the predictive abilities of such models to improve evaluating techniques, policymakers have gotten benefits from experiences in market areas, and shoppers have gone with additional proficient choices. The exploration findings introduced here act as a reason for future headway and very much educated decision-production in the field of feasible and proficient energy management, as the electric car industry keeps on developing.

REFERENCES

[1]. Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R. and Hinton, G.E., 2021. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, *34*, pp.4699-4711.

[2]. Al Mamari, A.H.S., Al Ghafri, R.S.H.H., Aravind, N., Dhandapani, R., Al Hatali, E.M.A.M. and Pandian, R., 2023. Experimental study and development of machine learning model using random forest classifier on shear strength prediction of RC beam with externally bonded GFRP composites. *Asian Journal of Civil Engineering*, *24*(1), pp.267-286.

[3]. Ayoub, A., Jia, Z., Szepesvari, C., Wang, M. and Yang, L., 2020, November. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning* (pp. 463-474). PMLR.

[4]. Bakay, M.S. and Ağbulut, Ü., 2021. Electricity production-based forecasting of greenhouse gas emissions in Turkey with deep learning, support vector machine, and artificial neural network algorithms. *Journal of Cleaner Production*, *285*, p.125324.

BULLET : Jurnal Multidisiplin Ilmu
Volume 4, No. 01, Februari - Maret 2025
ISSN 2829-2049 (media online)
Hal 99-118

[5]. Chandran, V., Patil, C.K., Karthick, A., Ganeshaperumal, D., Rahim, R. and Ghosh, A., 2021. State of charge estimation of lithium-ion battery for electric vehicles using machine learning algorithms. *World Electric Vehicle Journal*, *12*(1), p.38.

[6]. Coatrini-Soares, A., Coatrini-Soares, J., Neto, M.P., de Mello, S.S., Pinto, D.D.S.C., Carvalho, W.A., Gilmore, M.S., Piazzetta, M.H.O., Gobbi, A.L., de Mello Brandão, H. and Paulovich, F.V., 2023. Microfluidic E-tongue to diagnose bovine mastitis with milk samples using Machine learning with Decision Tree models. *Chemical Engineering Journal*, *451*, p.138523.

[7]. Hamdi, M., Hilali-Jaghdam, I., Elnaim, B.E. and Elhag, A.A., 2023. Forecasting and classification of new cases of COVID 19 before vaccination using decision trees and Gaussian mixture model. *Alexandria Engineering Journal*, *62*, pp.327-333.

[8]. Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N. and Peste, A., 2021. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *The Journal of Machine Learning Research*, *22*(1), pp.10882-11005.

[9]. Kishino, M., Matsumoto, K., Kobayashi, Y., Taguchi, R., Akamatsu, N. and Shishido, A., 2023. Fatigue life prediction of bending polymer films using random forest. *International Journal of Fatigue*, *166*, p.107230.

[10]. Liu, S., Wang, L., Zhang, W., He, Y. and Pijush, S., 2023. A comprehensive review of machine learning-based methods in landslide susceptibility mapping. *Geological Journal*.

[11]. Maulud, D. and Abdulazeez, A.M., 2020. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, *1*(4), pp.140-147.

[12]. Mazhar, T., Asif, R.N., Malik, M.A., Nadeem, M.A., Haq, I., Iqbal, M., Kamran, M. and Ashraf, S., 2023. Electric Vehicle Charging System in the Smart Grid Using Different Machine Learning Methods. *Sustainability*, *15*(3), p.2603.

[13]. Palimkar, P., Shaw, R.N. and Ghosh, A., 2022. Machine learning technique to prognosis diabetes disease: Random forest classifier approach. In *Advanced Computing and Intelligent Technologies: Proceedings of ICACIT 2021* (pp. 219-244). Springer Singapore.

[14]. Ruan, J., Wu, C., Liang, Z., Liu, K., Li, B., Li, W. and Li, T., 2023. The application of machine learning-based energy management strategy in a multi-mode plug-in hybrid electric vehicle, part II: Deep deterministic policy gradient algorithm design for electric mode. *Energy*, *269*, p.126792.

[15]. Tariq, A., Yan, J., Gagnon, A.S., Riaz Khan, M. and Mumtaz, F., 2023. Mapping of cropland, cropping patterns and crop types by combining optical remote sensing images with decision tree classifier and random forest. *Geo-Spatial Information Science*, *26*(3), pp.302-320.

[16]. Ullah, I., Liu, K., Yamamoto, T., Shafiullah, M. and Jamal, A., 2023. Grey wolf optimizer-based machine learning algorithm to predict electric vehicle charging duration time. *Transportation Letters*, *15*(8), pp.889-906.

[17]. Wang, Z., Hong, T. and Piette, M.A., 2020. Building thermal load prediction through shallow machine learning and deep learning. *Applied Energy*, *263*, p.114683.

[18]. Wu, C., Ruan, J., Cui, H., Zhang, B., Li, T. and Zhang, K., 2023. The application of machine learning based energy management strategy in multi-mode plug-in hybrid electric vehicle, part I: Twin Delayed Deep Deterministic Policy Gradient algorithm design for hybrid mode. *Energy*, *262*, p.125084.

[19]. Zermane, A., Tohir, M.Z.M., Zermane, H., Baharudin, M.R. and Yusoff, H.M., 2023. Predicting fatal fall from heights accidents using random forest classification machine learning model. *Safety science*, *159*, p.106023.

[20]. Zhao, J., Ling, H., Liu, J., Wang, J., Burke, A.F. and Lian, Y., 2023. Machine learning for predicting battery capacity for electric vehicles. *ETransportation*, *15*, p.100214.

[21]. Zhang, B., Tian, J., Pei, S., Chen, Y., He, X., Dong, Y., Zhang, L., Mo, X., Huang, W., Cong, S. and Zhang, S., 2019. Machine learning–assisted system for thyroid nodule diagnosis. *Thyroid*, *29*(6), pp.858-867.

[22]. [22] Li, G., Zhou, X. and Cao, L., 2021, October. Machine learning for databases. In *Proceedings of the First International Conference on AI-ML Systems* (pp. 1-2).

[23]. Wang, Y., Liu, M., Yang, J. and Gui, G., 2019. Data-driven deep learning for automatic modulation recognition in cognitive radios. *IEEE Transactions on Vehicular Technology*, *68*(4), pp.4074-4077.

[24]. Hasan, M., Al Sany, S. A., & Swarnali, S. H. (2024). HARNESSING BIG DATA AND MACHINE LEARNING FOR TRANSFORMATIVE HEALTHCARE INFORMATION MANAGEMENT. *Unique Endeavor in Business & Social Sciences*, *3*(1), 231-245.