Car Prices Estimation Project report

Student: Elena Zdravkoska
Faculty: UP FAMNIT

1 Preface

My Car Prices Estimation Project is a Data Mining and Machine Learning project which focuses on developing a predictive model to estimate car prices based on variety of features , using a dataset that I have obtained from Kaggle. With the use of two different machine learning approaches: Linear Regression and Neural networks, I tried to capture the linear and non-linear aspects of the data. The dataset contains key features such as car make, model, manufacturing year, mileage and condition. Using comparison between the models, in the project I aim to determine the most effective way to predict the car price.

2 About the dataset

I used the “Car Price Prediction Dataset” which contains a .csv file with 4009 rows and 10 columns.
It has various car brands which contain some key features as:

->Make: Brand of the car which is categorical and influences the prices based on the brand quality and reputation.

->Model: Specific model of a brand which is also categorical affecting the value of the car based on specific features.

->Manufacturing year: Manufacturing year of the car which is numerical and affects the price in a way if the car model is newer generally means that the price will be higher.

->Mileage: Distance that the car has traveled which is numerical and influences the price in a way if the car has passes more km the price will be lower.

->Condition: Overall state of the car which is again categorical and impacts the value of the car directly based on appearance and maintenance.

->Price: Market price of the car which is numerical and also our target variable that the models aim to predict.

	Unnamed: 0	Make	Model	Year	Mileage	Condition	Price
0	0	Ford	Silverado	2022	18107	Excellent	19094.75
1	1	Toyota	Silverado	2014	13578	Excellent	27321.10
2	2	Chevrolet	Civic	2016	46054	Good	23697.30
3	3	Ford	Civic	2022	34981	Excellent	18251.05
4	4	Chevrolet	Civic	2019	63565	Excellent	19821.85

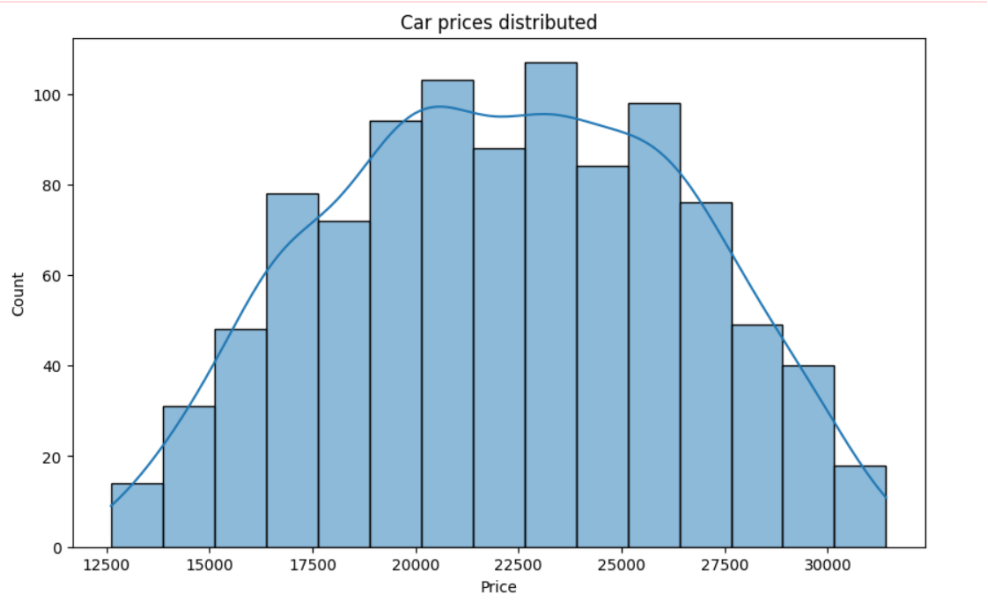
The dataset that I chose as we can see is a blend of numerical and categorical features, each very important for determining the car price.

3 Approach

My approach includes data preprocessing, model implementation, and evaluation. Two modeling approaches were undertaken: Linear Regression as a baseline and Neural Networks for advanced predictive capabilities.

3.1 Visualization od the distributed car prices

A histogram which shows us how the car prices are distributed.



3.2 Data preprocessing

I used the data preprocessing considering that I have numerical and categorical features.
I have used several steps to make sure that the data is prepared:

->Handling missing values: I have started off with handling the missing values using dropna() it is very important to make sure that I have a clean dataset which is ready for modeling. The numerical features were replaced with the average to maintain overall balance and the categorical were replaced with the most common value to keep typical patterns.

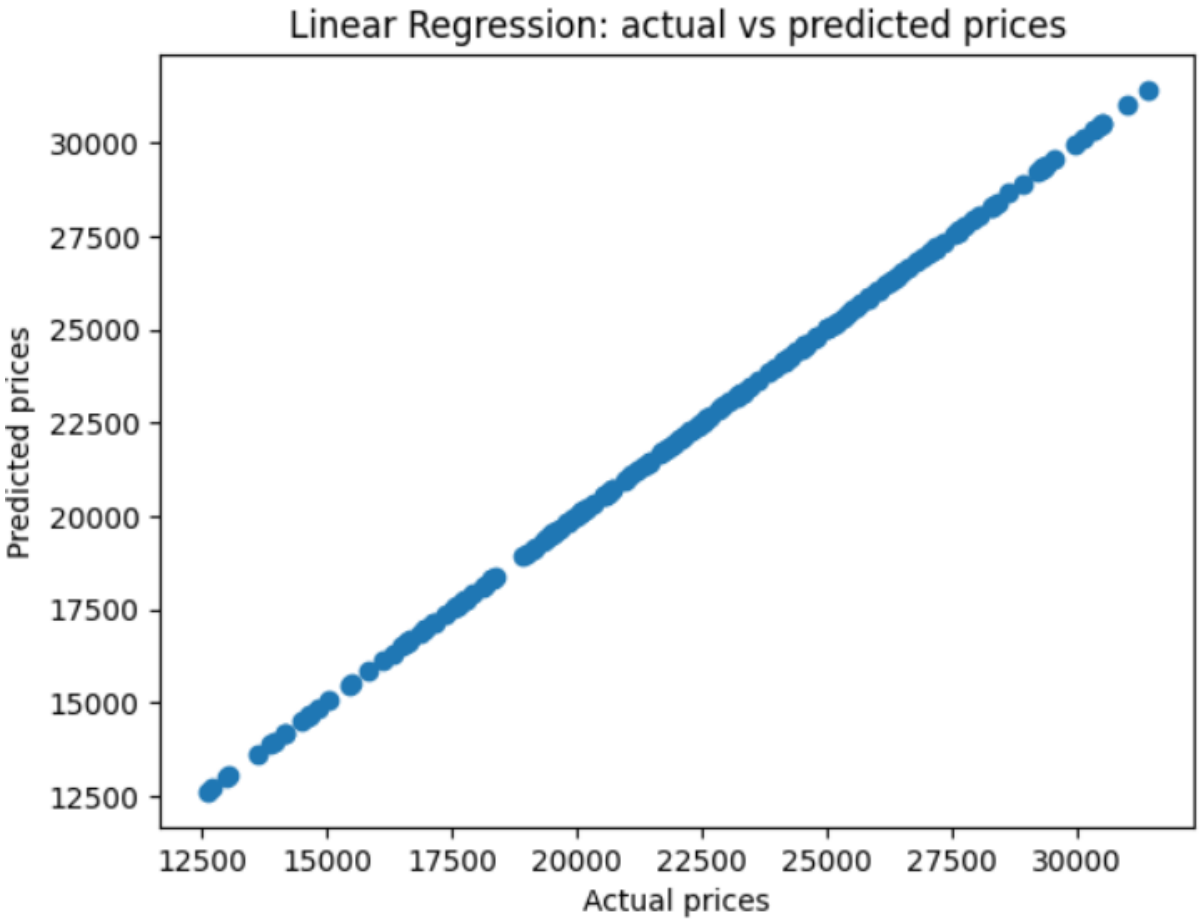
->Categorical encoding: I converted the categorical features into numeric representations using one-hot encoding via get_dummies(). This is crucial because machine learning models require numeric inputs.

-> Feature and Target Separation: I split the dataset into features (X) and the target variable (y). Here, X consists of all features except the price, which is stored in y.

3.3 Linear Regression

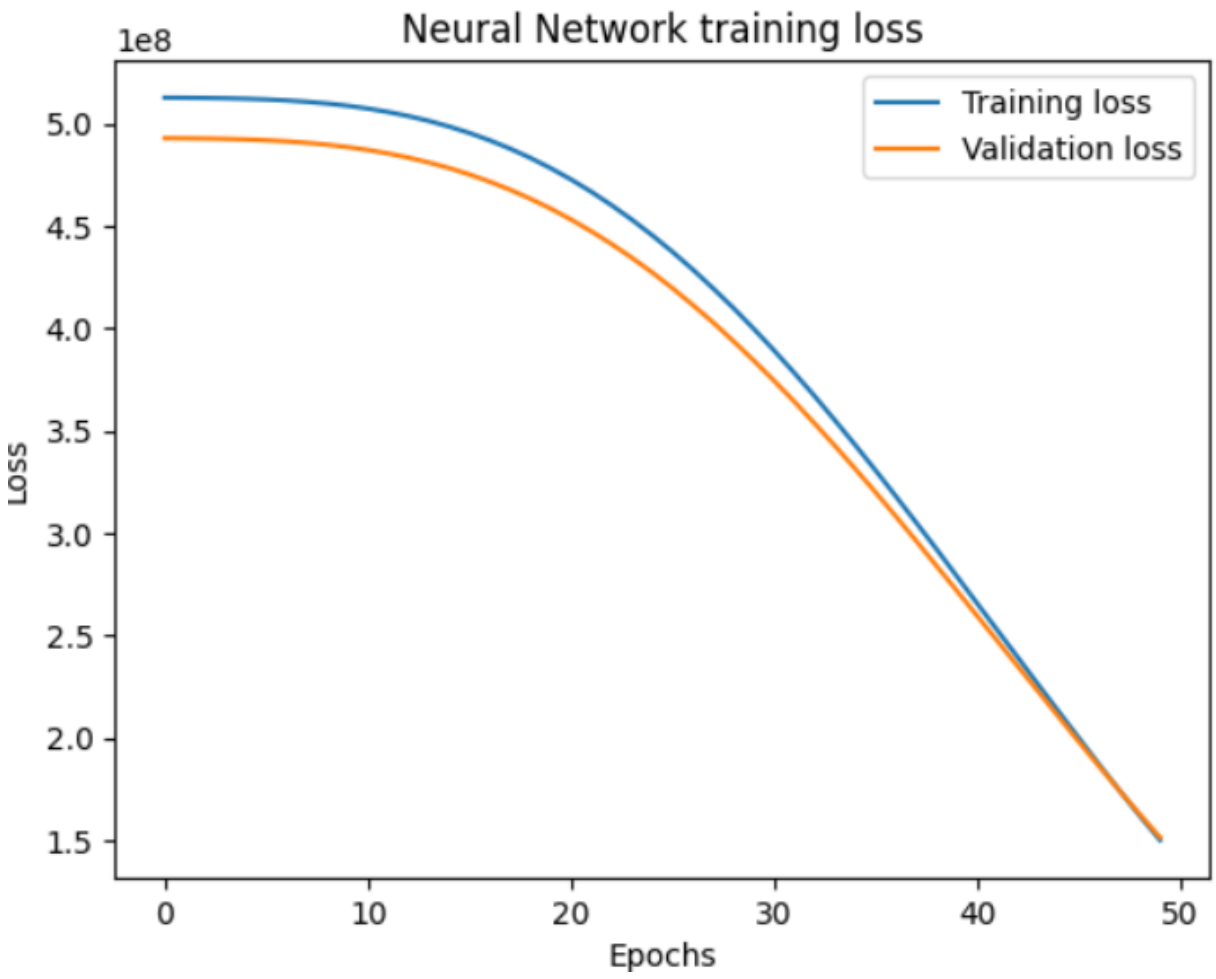
As my baseline model I have used Linear Regression due to its simplicity. The model assumes a linear relationship between the features of the cars and their prices and was initialized and evaluated using cross-validation to assess its performance on different training subsets. The data was split into features (X) and the target variable (y), using an 80-20 ratio. This allowed the model to be trained on one portion of the data and evaluated on another to assess its generalization capability. I computed the Mean Squared Error (MSE) for each fold and averaged them to gauge the model's error. After training on the full training set, the model's predictions on the test set were assessed using MSE, Mean Absolute Error (MAE), and R-squared metrics. Additionally, a scatter plot of actual versus predicted prices was created to visualize prediction accuracy and identify any potential biases.

```
Linear regression Mean Squared Error: 0.0047774769462206835
Linear regression Cross-Validation MSE: 0.004296551633736129
Linear regression Mean Squared Error: 0.0047774769462206835
Linear regression Mean Absolute Error: 0.06002836477567144
Linear regression R^2 Score: 0.9999999997642285
```



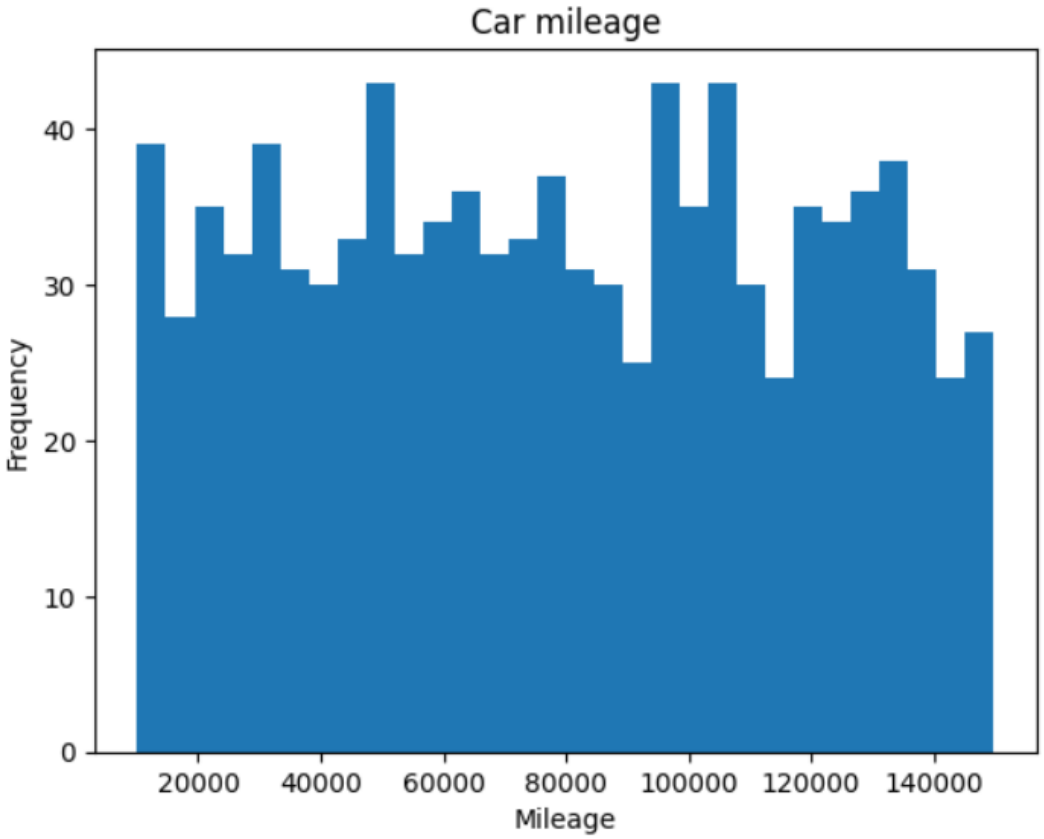
3.4 Neural Networks

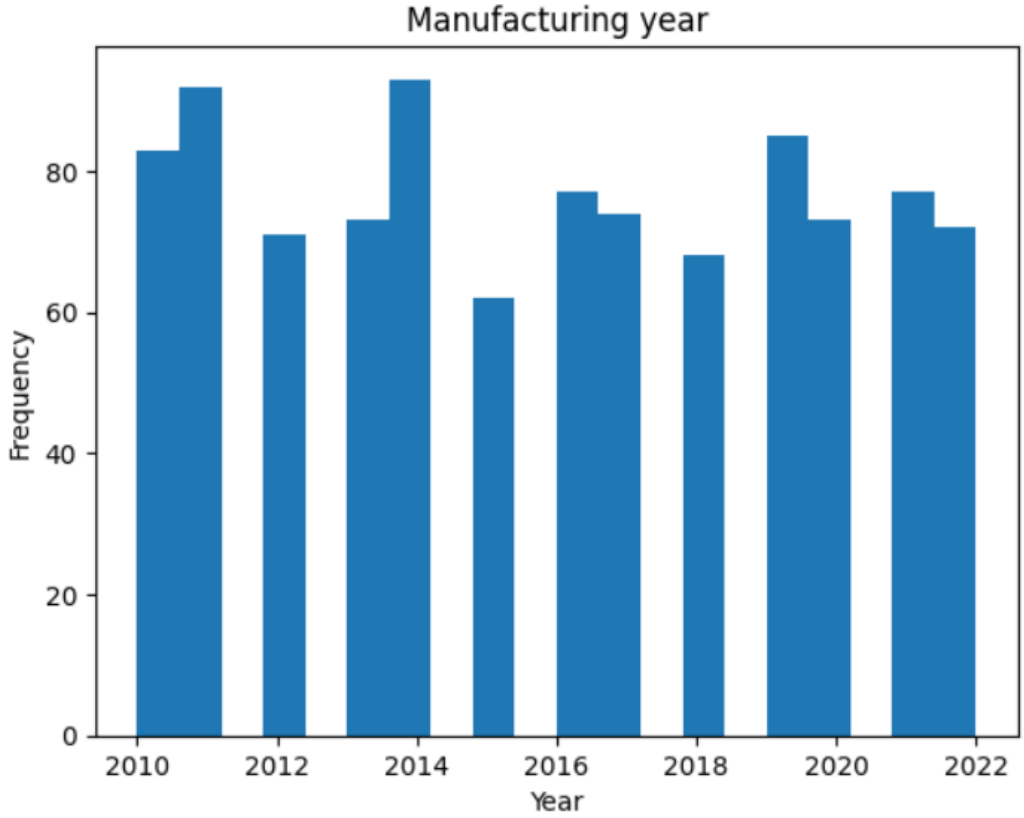
As my advanced model I have used Neural Networks due to the fact that they are proficient at modeling non-linear interactions and capturing intricate patterns within data. To prepare the data for the model I normalized the features using StandartScaler to achieve a mean of zero, so the training would be effective. The model was build using TensorFlow and Keras, features an input layer, two hidden layers with ReLU activation for capturing non-linear relationships, and a linear output layer for predicting car prices. I trained the model for 50 epochs, the network used the Adam optimizer to minimize Mean Squared Error (MSE), with training and validation losses monitored to avoid overfitting or underfitting.

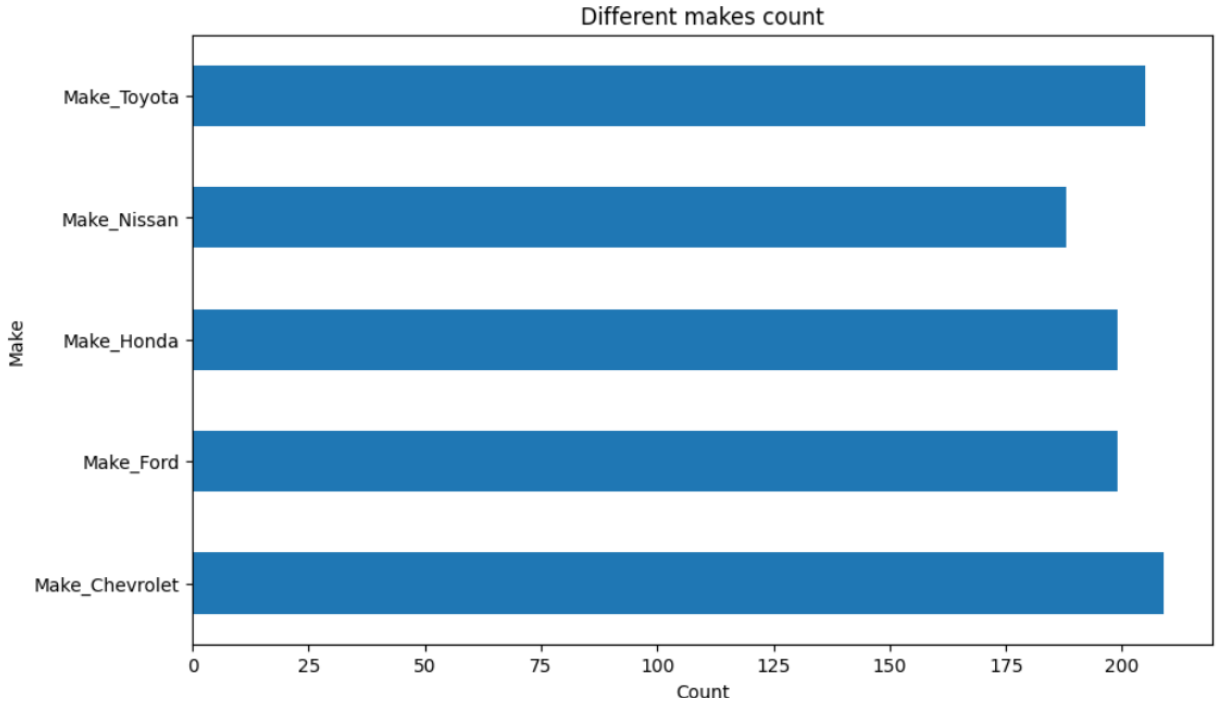


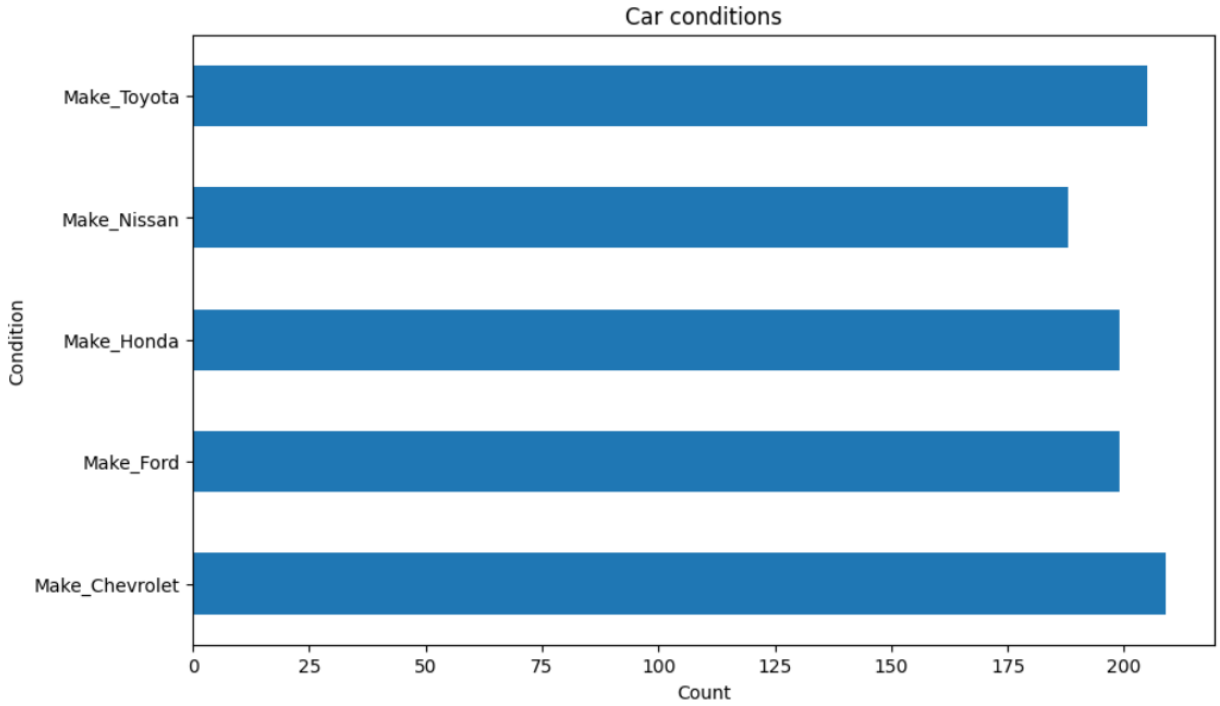
3.5 Histograms

I have used histograms to visualize the key car features: mileage, manufacturing year, make, and condition. The mileage histogram shows the distribution of mileage, which highlights the common ranges and outliers. The manufacturing year histogram reveals the age distribution of the cars. Bar charts of car makes and conditions provide insights into the frequency of each make and condition, helping me to understand their distribution and potential impact on pricing. These histograms are of great importance to gras the needed dataset's characteristics and to guide me through the model development.


Car mileage
Frequency
0
10
20
30
40
20000
40000
60000
80000
100000
120000
140000
Mileage


Manufacturing year
Frequency
0
20
40
60
80
2010
2012
2014
2016
2018
2020
2022
Year


Different makes count
Make_Toyota
Make_Nissan
Make_Honda
Make_Ford
Make_Chevrolet
Make
0
25
50
75
100
125
150
175
200
Count


Car conditions
Make_Toyota
Make_Nissan
Make_Honda
Make_Ford
Make_Chevrolet
Condition
0
25
50
75
100
125
150
175
200
Count

4 Results

In evaluating the performance of the two models, I used Mean Squared Error (MSE) and R-squared (R^2) metrics. For the Linear Regression model, I observed an R^2 of 0.9999999997642285 on the training set and an MSE of 0.0047774769462206835 on the test set, indicating exceptionally accurate predictions with very low error. In contrast, the Neural Network model, which does not typically use R^2 for evaluation, achieved an MSE of 149484553.65045938 on the test set. This highlights the Neural Network's ability to handle complex patterns, though its higher MSE indicates that it may be less accurate in prediction compared to the Linear Regression model. Overall, while the Neural Network captures intricate relationships, the Linear Regression model demonstrated outstanding performance with minimal error.

5 Conclusion

In the project I successfully demonstrated the application of Linear Regression and Neural Networks in predicting car prices. Future work I would like to could explore the integration of additional features, like hyperparameter tuning through GridSearchCV to optimize the neural network's performance.

6 References

Kaggle Dataset: Source of the dataset that I have used.
https://www.kaggle.com/datasets/mrsimple07/car-prices-prediction-data

Matplotlib Documentation: Reference for plotting and visualization.

Scikit-learn Documentation: Reference for implementing Linear Regression, preprocessing, and creating model pipelines.

TensorFlow/Keras Documentation: Reference for building and training the Neural Network model.

Pandas Documentation: Reference for data manipulation and preprocessing tasks.