EE363 Minor-1 Project- Review Presentation

Car Price Prediction Using Linear Regression
Machine Learning Project

- **NATIONAL INSTITUTE OF TECHNOLOGY ANDHRA PRADESH**
- **Department of Electrical and Electronics Engineering**

Group (single)
Shubham Kumar
Roll no.-521242
Sec-'B'

Faculty supervisor
Dr. K. Sri Phani Krishna
Assistant Professor

Title: Car Price Prediction Using Linear Regression

Introduction

- What is Linear Regression?

- Linear regression is a supervised machine learning algorithm.

- It is used to model the relationship between a dependent variable (y) and one or more independent variables (X).

- The goal of linear regression is to find a linear equation that best fits the data points.

- The equation is of the form: y = mx + b, where:
- m is the slope of the line
- b is the y-intercept

Data Collection

- The first step in building a car price prediction model is to collect data.
- This data can be collected from a variety of sources, such as online car listings, car dealerships, or government agencies.
- The data should include the following information for each car:
- Price
- name
- company
- Year
- kms_driven
- fuel_type

- Data set is collected from quicker.com website

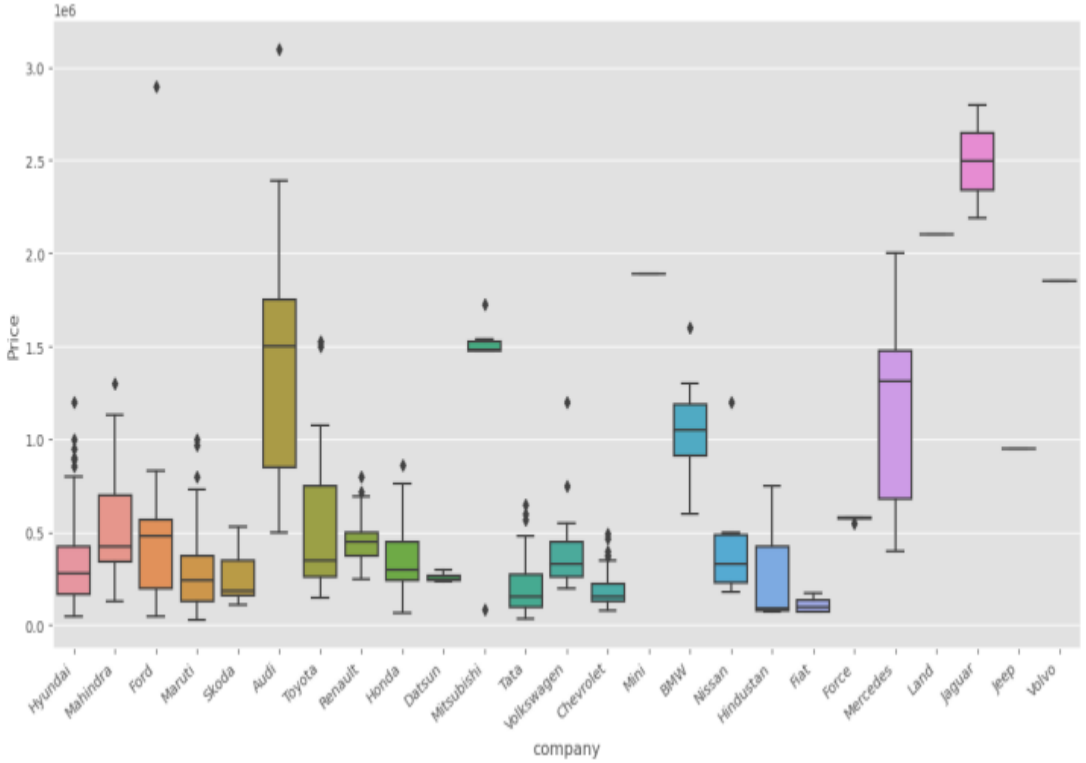
Data Preprocessing

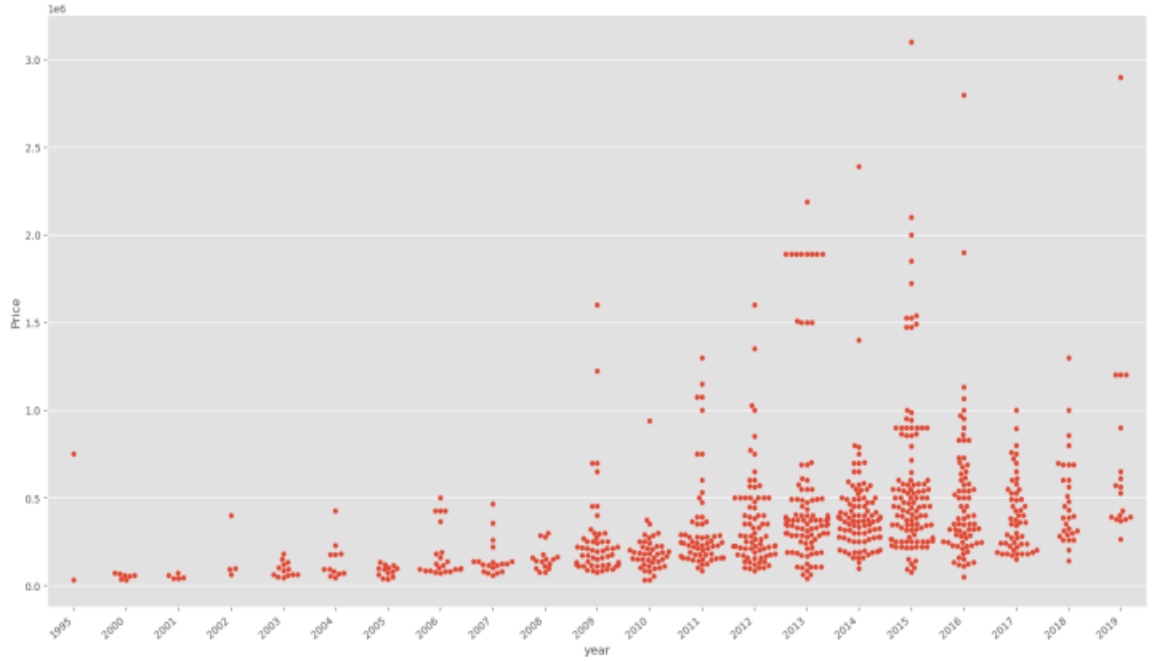
- Once you have collected your data, you will need to preprocess it before you can use it for training your model.
- Data preprocessing can include the following steps:
- Handling missing values: Missing values can be imputed or removed from the data.
- Dealing with outliers: Outliers can be removed from the data or winsorized.
- Encoding categorical variables: Categorical variables need to be encoded into numerical values before they can be used in a linear regression model.
- Scaling numerical variables: Numerical variables may need to be scaled to the same range to improve the performance of the model.

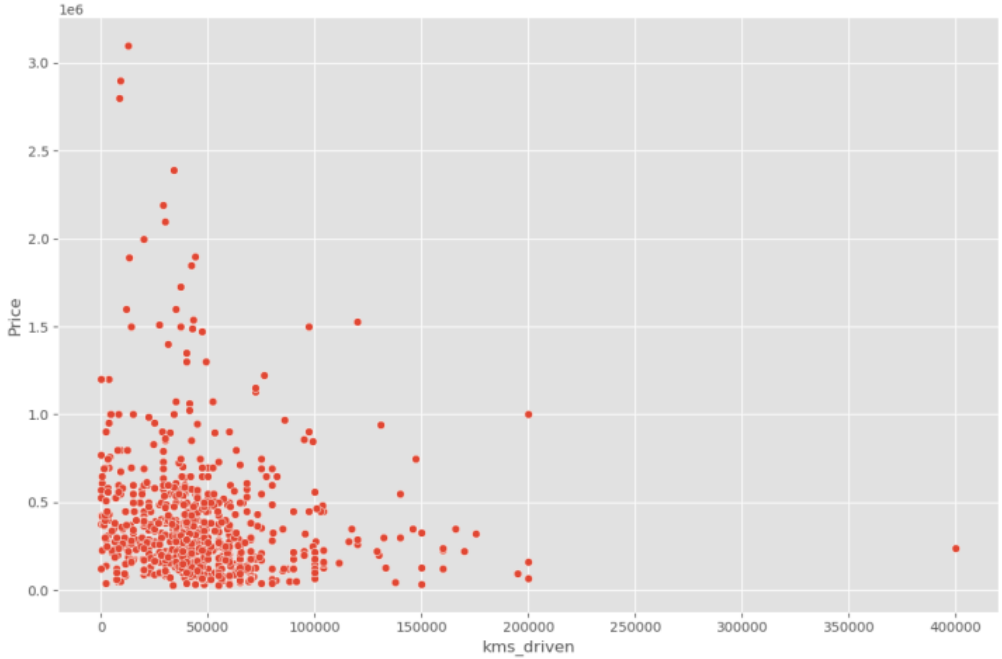
	name	company	year	kms_driven	fuel_type
0	Hyundai Santro Xing	Hyundai	2007	45000	Petrol
1	Mahindra Jeep CL550	Mahindra	2006	40	Diesel
2	Hyundai Grand i10	Hyundai	2014	28000	Petrol
3	Ford EcoSport Titanium	Ford	2014	36000	Diesel
4	Ford Figo	Ford	2012	41000	Diesel
...	...	...	...	...	...
811	Maruti Suzuki Ritz	Maruti	2011	50000	Petrol
812	Tata Indica V2	Tata	2009	30000	Diesel
813	Toyota Corolla Altis	Toyota	2009	132000	Petrol
814	Tata Zest XM	Tata	2018	27000	Diesel
815	Mahindra Quanto C8	Mahindra	2013	40000	Diesel

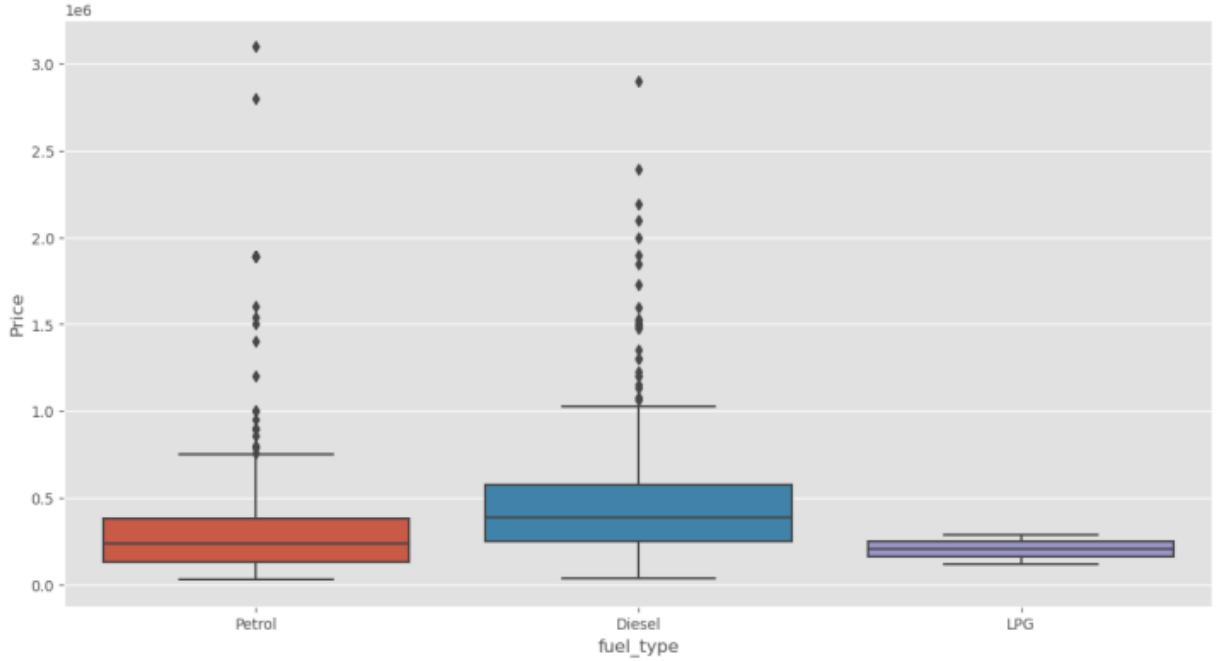
Cleaned data

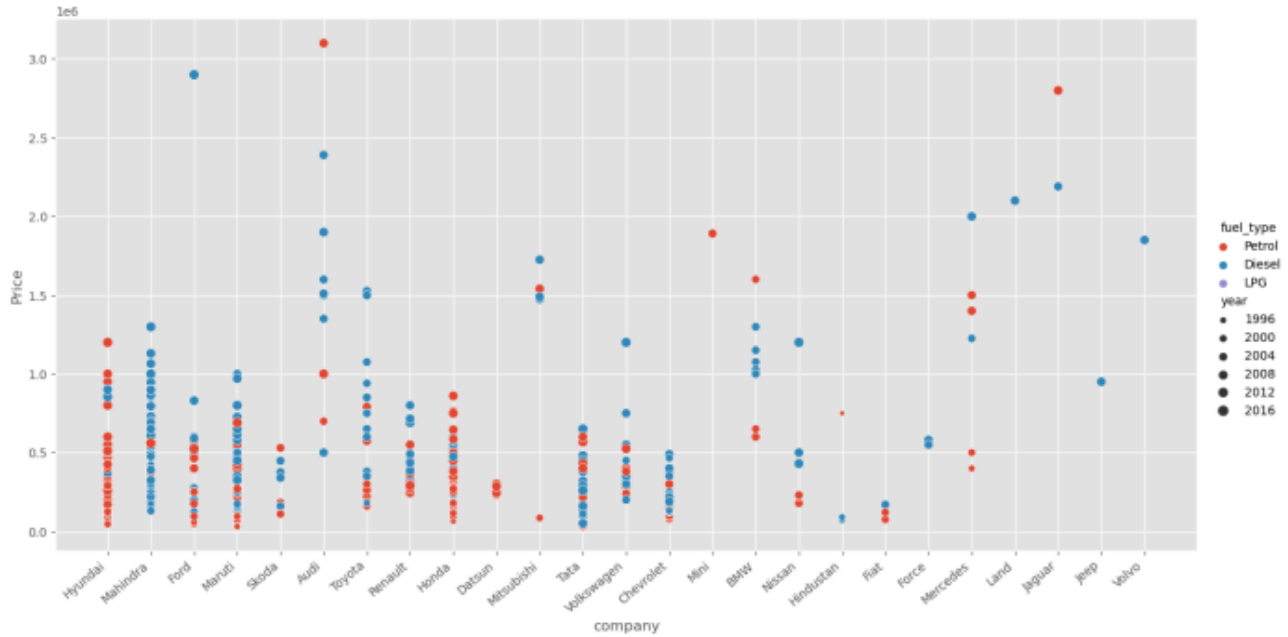
Graphs of relations between features





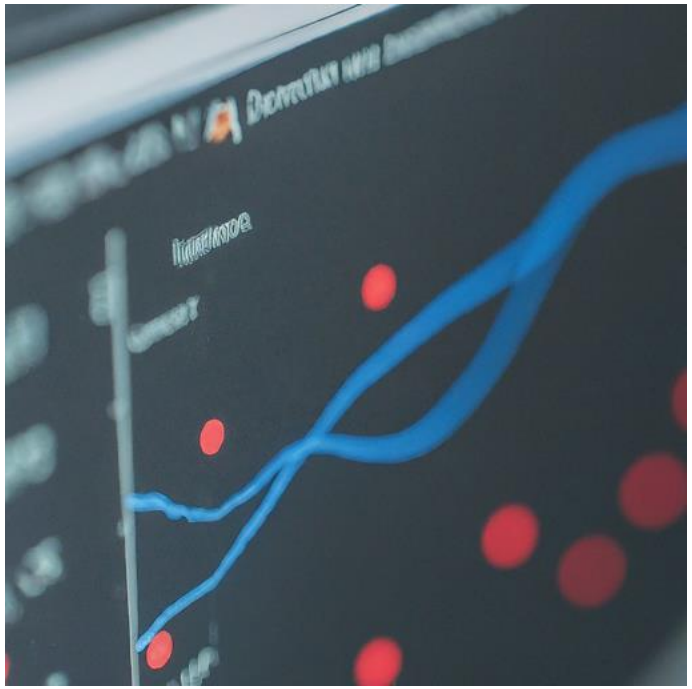

1e6
Price
kms_driven
0
50000
100000
150000
200000
250000
300000
350000
400000


1e6
Price
Petrol
Diesel
LPG
fuel_type


1e6
Price
company
Hyundai
Mahindra
Ford
Maruti
Skoda
Audi
Toyota
Renault
Honda
Datsun
Mitsubishi
Tata
Volkswagen
Chevrolet
Mini
BMW
Nissan
Hindustan
Fiat
Force
Mercedes
Land
Jaguar
Jeep
Volvo
fuel_type
Petrol
Diesel
LPG
year
1996
2000
2004
2008
2012
2016

Training the Model

- Splitting Data: Use the train_test_split function to split the dataset X and target variable y into training and testing sets. The test_size parameter specifies the proportion of the dataset to include in the test split.
- One-Hot Encoding: Instantiate a OneHotEncoder object (ohe) to encode categorical features. Then, fit the encoder on the categorical features (['name', 'company', 'fuel_type']) in the training data (X[['name','company','fuel_type']]) to learn the categories.
- Column Transformation: Create a ColumnTransformer object (column_trans) to apply transformations selectively to different columns in the dataset. In this case, you're applying one-hot encoding to the categorical features and passing through the remaining features unchanged.
- Instantiate Model: Instantiate a LinearRegression object (lr) to serve as the predictive model.
- Pipeline Creation: Create a pipeline (pipe) using make_pipeline. The pipeline will sequentially apply the transformations defined in column_trans and then fit the linear regression model (lr) to the transformed data.
- Model Training: Fit the pipeline (pipe) to the training data (X_train, y_train). This involves applying the transformations defined in column_trans to the training data and then fitting the linear regression model to the transformed data.



Evaluation

- Once you have trained your model, you need to evaluate its performance.
- Here are some common metrics used to evaluate linear regression models:
- R-squared (R^2): R^2 measures the proportion of variance in the dependent variable that is explained by the independent variables.

```
r2_score(y_test,y_pred)
```

```
0.6554468663859585
```

Prediction

- Once you have trained and evaluated your model, you can use it to predict the price of a new car.
- To predict the price of a new car, you simply need to input the car's features into the model.
- The model will then output a predicted price for the car.

```
pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti Suzuki Swift','Maruti',2019,100,'Petrol']).reshape(1,5)))
```

```
array([430315.44301096])
```

Conclusion

- Linear regression is a powerful tool that can be used to predict car prices.
- By understanding the relationship between car features and price, we can build models that can make accurate predictions.
- Car price prediction models can be used for a variety of purposes, such as helping consumers and businesses make informed decisions.