A Presentation on Car Price Prediction

Submitted To: Khusboo Garg (SME of Internship Batch no-28)

Submitted By: Abhishek Behera
Internship Batch no-28

Content

- Introduction & Problem Statement
- Literature Review
- Objective
- Analytical Problem Framing
- Analytical Problem Framing
- Data Sources and their formats
- Data Pre-processing
- Data Inputs- Logic- Output Relationships
- Model/s Development and Evaluation
- Visualizations
- Result Interpretation
- Conclusion & future Scope
- References:

PROBLEM STATEMENT

With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make car price valuation model.

DATA COLLECTION PHASE

- we have to scrape at least 5000 used cars data. You can scrape more data as well, it's up to you. more the data better the model.

- In this section You need to scrape the data of used cars from websites (OLX, Car Dekho, Cars 24 etc.) You need web scraping for this. You have to fetch data for different locations. The number of columns for data doesn't have limit, it's up to you and your creativity. Generally, these columns are Brand, model, variant, manufacturing year, driven kilometers, fuel, number of owners, location and at last target variable Price of the car. This data is to give you a hint about important variables in used car model. You can make changes to it, you can add or you can remove some columns, it completely depends on the website from which you are fetching the data.

- I have tried to include all types of cars in your data for example- SUV, Sedans, Coupe, minivan, Hatchback.

MODEL BUILDING PHASE

After collecting the data, you need to build a machine learning model. Before model building do all data pre-processing steps. Try different models with different hyper parameters and select the best model. Follow the complete life cycle of data science. Include all the steps below.

1. Data Cleaning

2. Exploratory Data Analysis

3. Data Pre-processing

4. Model Building

5. Model Evaluation

6. Selecting the best model

DATA SCIENCE LIFE CYCLE

Data Cleaning

- Import the collected data from web scraping
- Clean and format the records as per usage by using various imputation techniques

Exploratory Data Analysis

- Check through all the dataset information like datatype, missing value, duplicate value etc.
- Analyze each and every data record to ensure we have usable information

Visualization and Data Preprocessing

- Use various visualization methods to check the data distribution identify presence of outliers and skewness
- Perform encoding and scaling methods

DATA SCIENCE LIFE CYCLE

Model Building

- Create appropriate Regression Machine Learning model function
- Need to ensure that whenever the regression function is called it is able to process all the necessary parameters

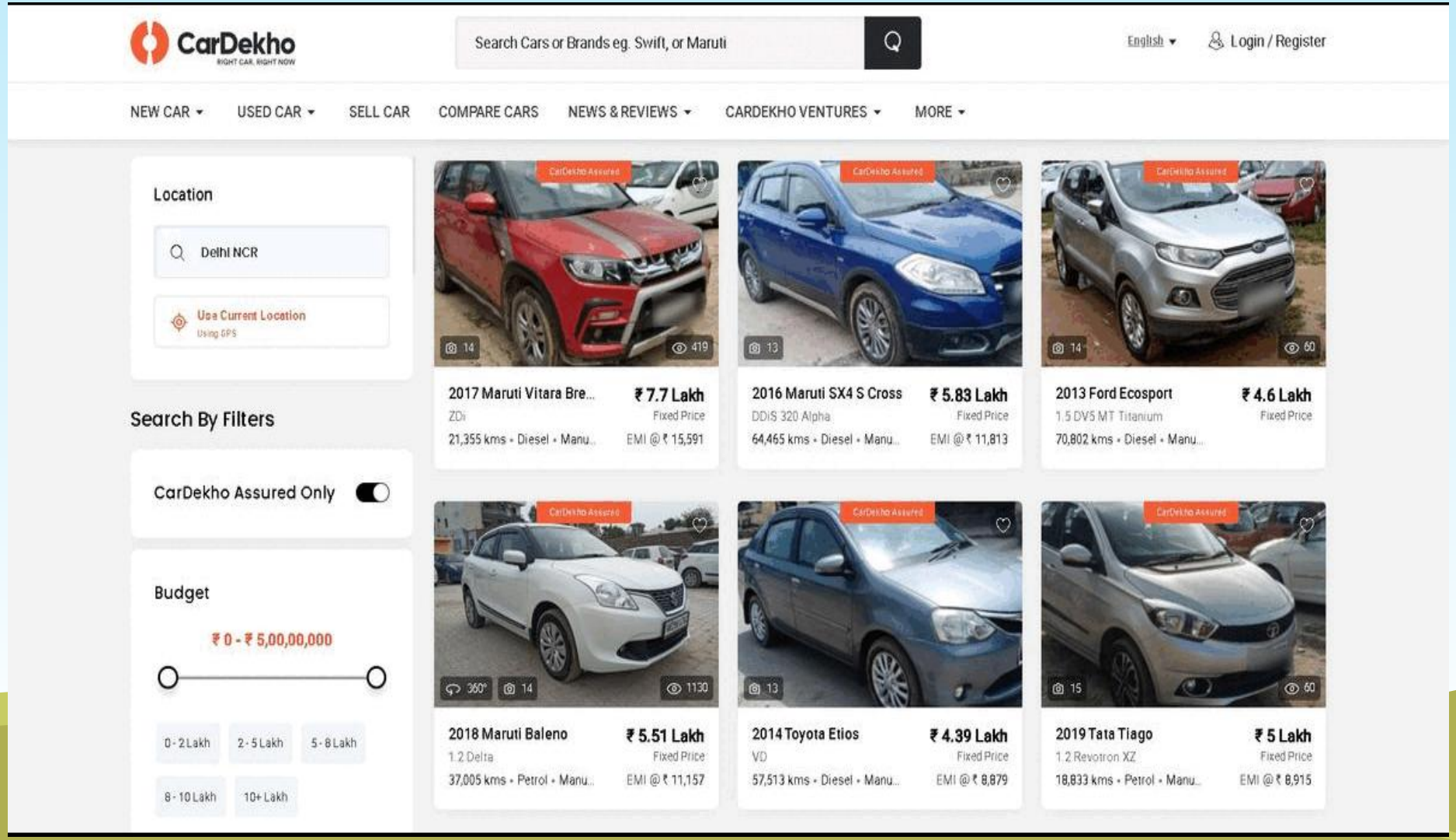
Model Evaluation

- Usage of evaluation metrics to check the accuracy of the models over trained and test data inputs
- Ensure the cross validation techniques helps in reducing over fitting and under fitting data

Hyperparameter Tuning Best Model

- Choosing the appropriate Regression Machine Learning model to check various parameter permutation and combinations
- Using Grid Search CV to obtain the best parameters that can be plugged into the selected model

WEB SCRAPING FOR USED CAR DETAILS



DATA PREPROCESSING

- Importing the necessary dependencies and libraries.
- Reading the CSV file and converted into data frame.
- Checking the data dimensions for the original dataset.
- Looking for null values and accordingly fill the missing data.
- Checking the summary of the dataset.
- Checking unique values.
- Checking all the categorical columns in the dataset.

DATA PREPROCESSING

- Visualizing each features using matplotlib and seaborn.
- Performing encoding using the ordinal encoder on categorical features.
- Checking for co-relation/multi-collinearity in a heatmap.
- Checking for Outliers/Skewness using boxen plot and distribution plot.
- Perform Scaling using Standard Scaler method.
- Checking for the final dimension of dataset to confirm the input details.
- Creating train test split and the best random state found in the range 1-1000.

TECHNOLOGY USED

- Hardware technology being used.

RAM : 8 GB

- CPU : Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
- GPU : Intel(R) HD Graphics 5500 and NVIDIA GeForce 940M

- Software technology being used.

Programming language : Python

Distribution : Anaconda Navigator

Browser based language shell : Jupyter Notebook

- Libraries/Packages specifically being used.

Pandas, NumPy, matplotlib, seaborn, scikit-learn, pandas-profiling, missingno

EXPLORATORY DATA ANALYSIS (EDA) AND VISUALIZATION

01. Univariate Analysis

Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable.

02. Multivariate Analysis

Multivariate analysis is a set of statistical techniques used for **analysis** of data that contain more than one variable.

03. Correlation of Dataset

Correlation is used to test relationships between quantitative variables or categorical variables.

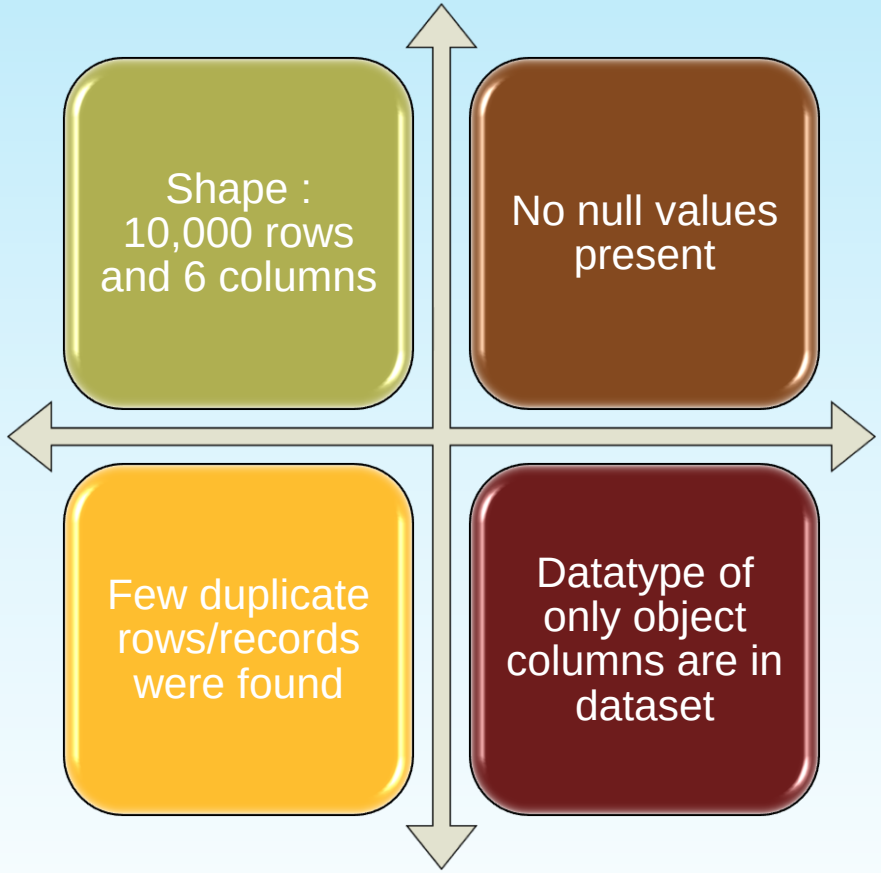
04. Correlation with Target variable

Correlation with the target variable to know how the data is related.

05. Conclusion

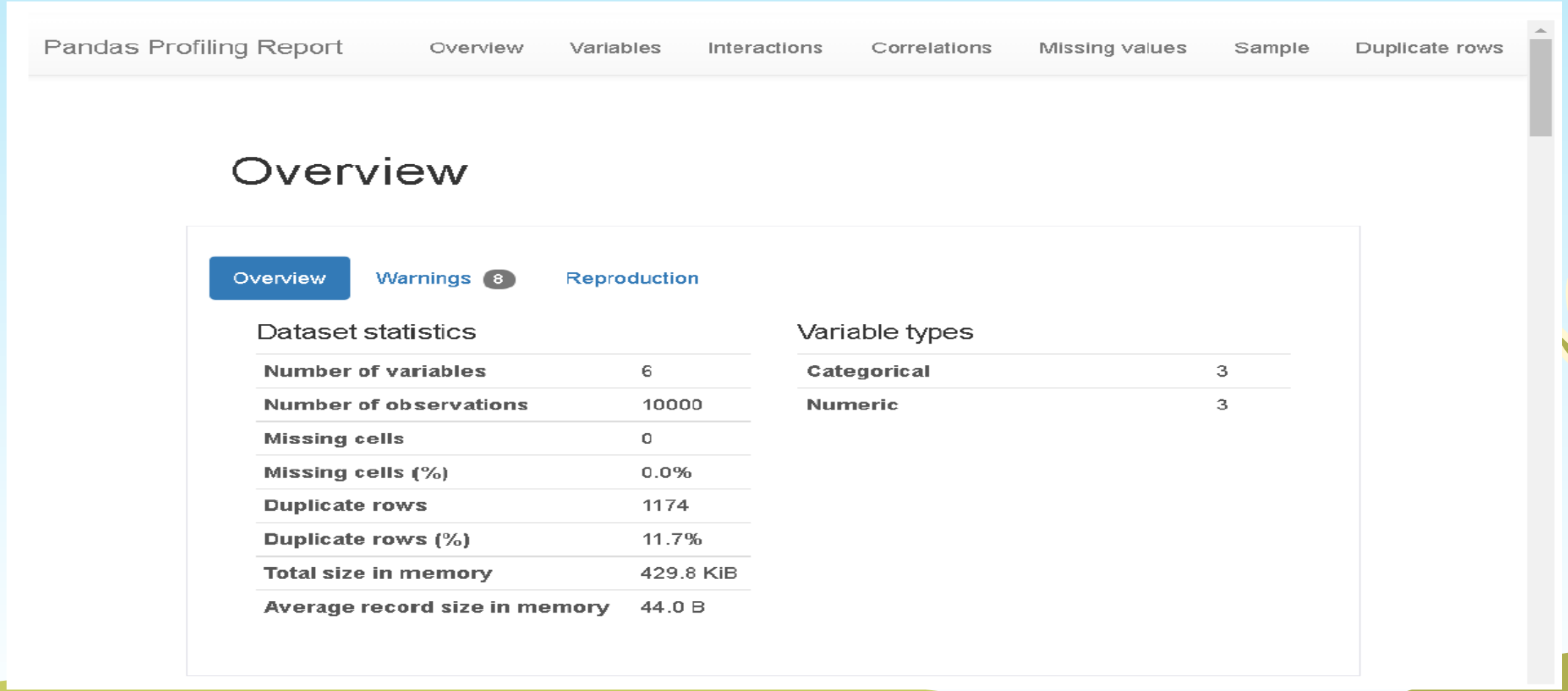
Summary with the conclusion of all the analysis

EXPLORATORY DATA ANALYSIS (EDA)

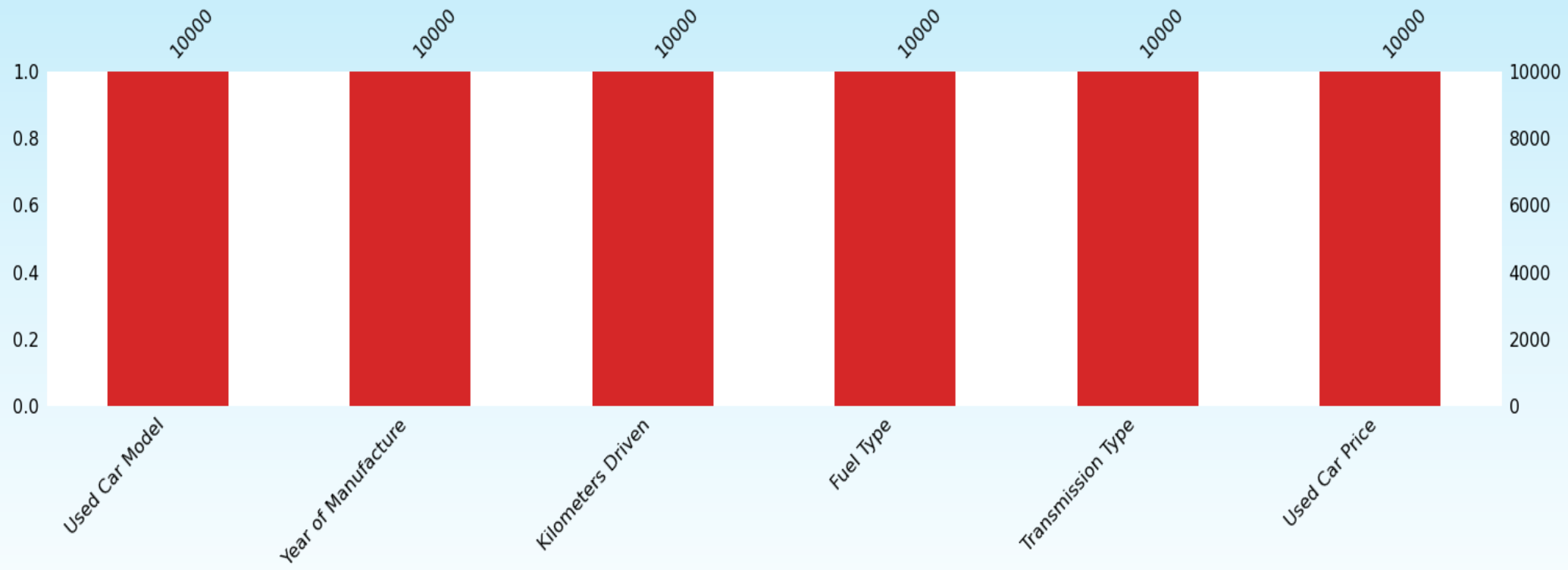


- First I have imported the necessary libraries and loaded the entire dataset in our Jupyter Notebook and renamed the project file from untitled.
- Then I checked the shape of our dataset and found that we have a total of 10,000 rows and 6 different columns.
- We don’t have any null values or missing values present in our dataset from the web scraping.
- There few duplicate rows/records in our dataset but I decided to retain them instead of deleting it.
- By checking the data types I came to know that our data set consists of columns having only object datatype even those there were numeric information present.

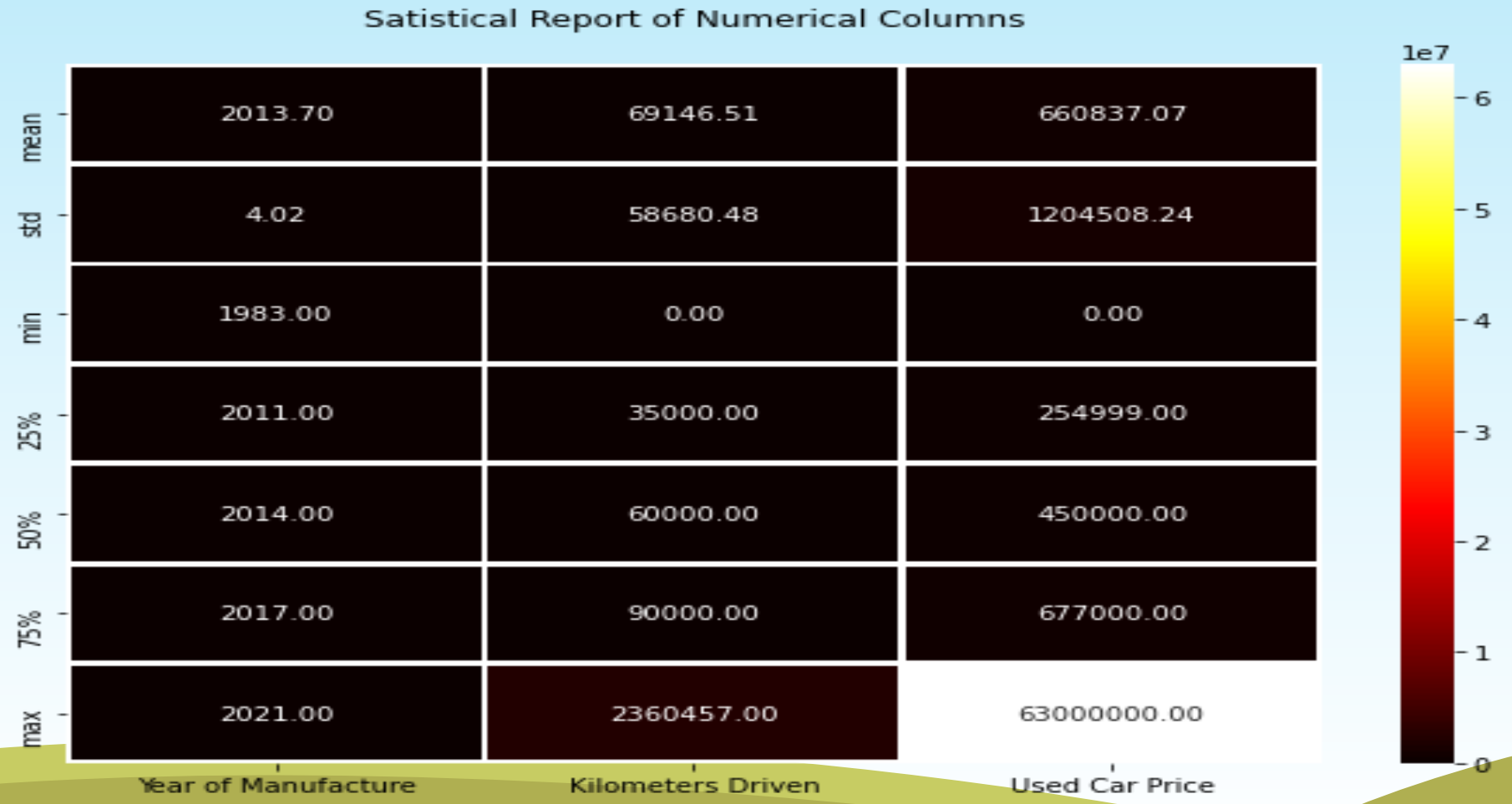
VISUALIZATION USING PANDAS PROFILING REPORT



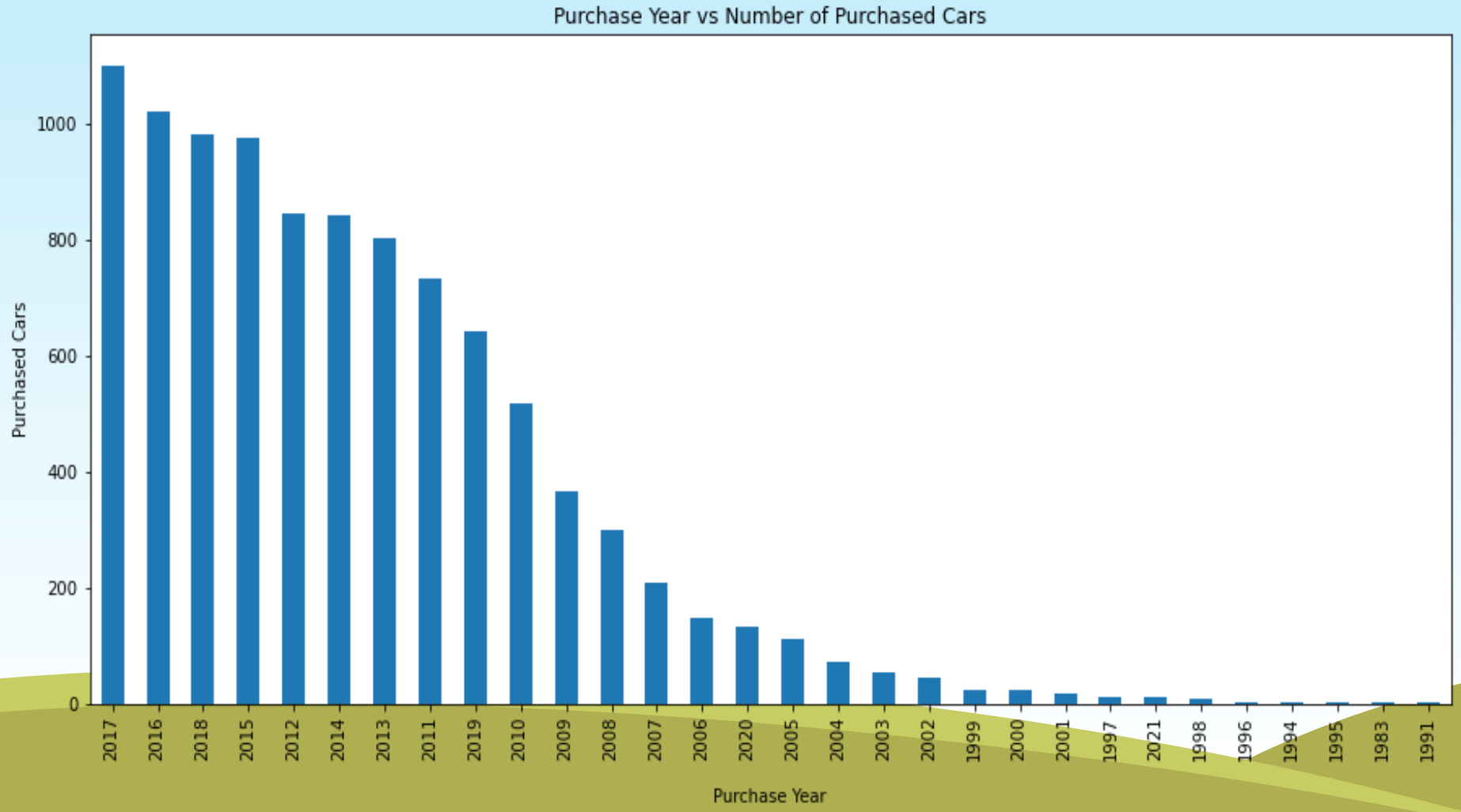
MISSING VALUES VISUAL USING MISSINGNO



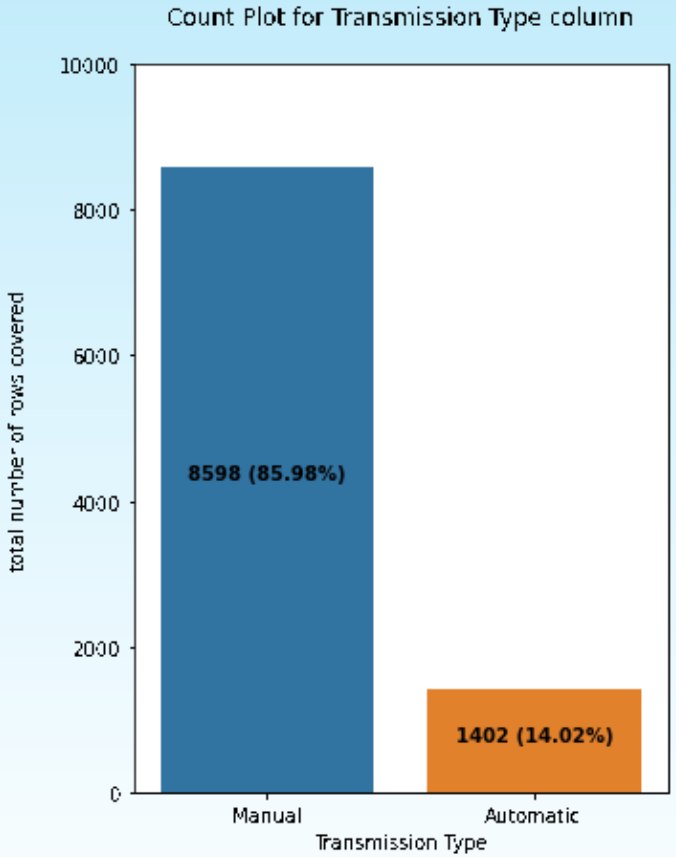
DESCRIBE DATASET VISUAL ON NUMERIC DATA



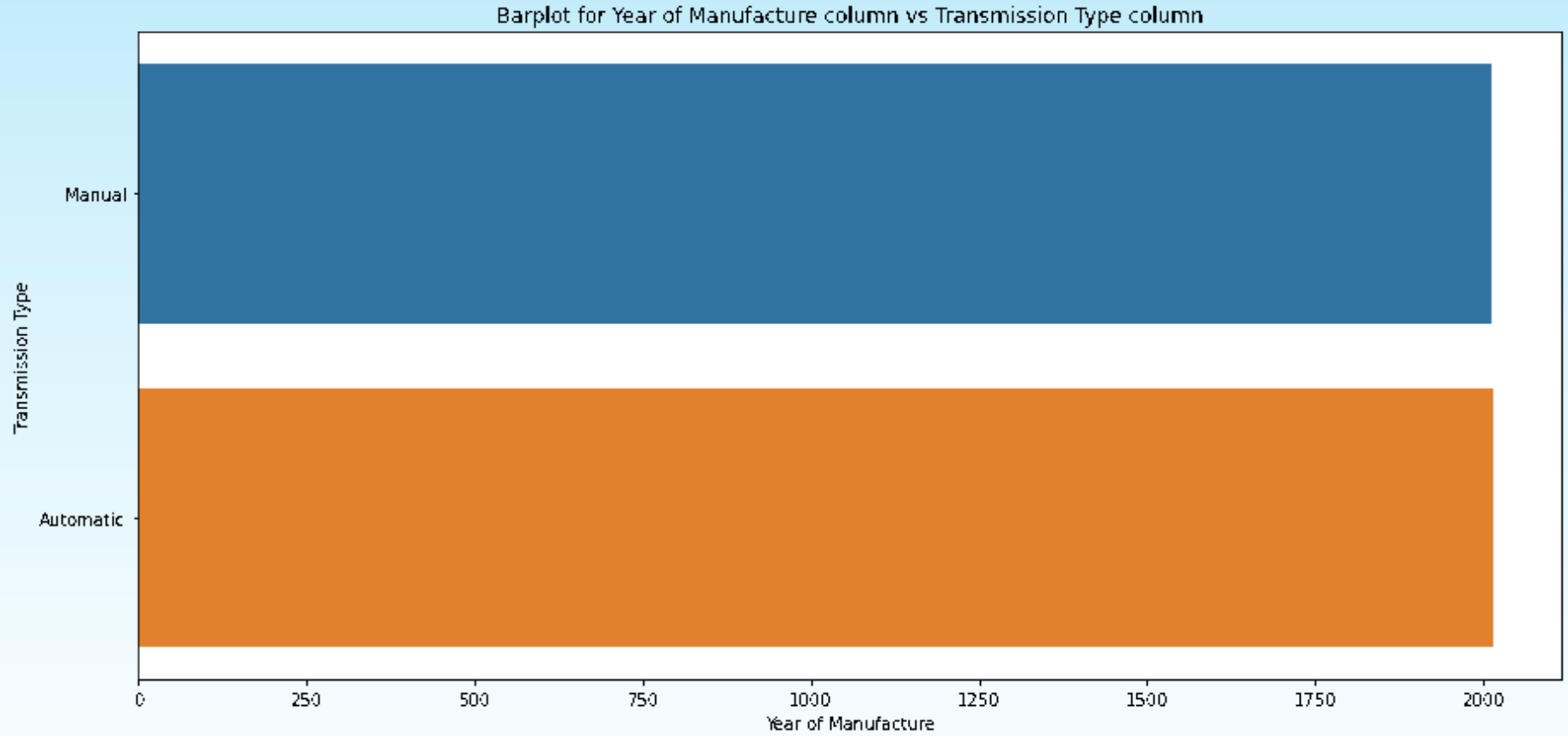
PURCHASE DETAILS OF USED CARS EACH YEAR



COUNT PLOTS


Count Plot for Transmission Type column
10000
8000
6000
4000
2000
0
total number of rows covered
8598 (85.98%)
1402 (14.02%)
Manual
Automatic
Transmission Type

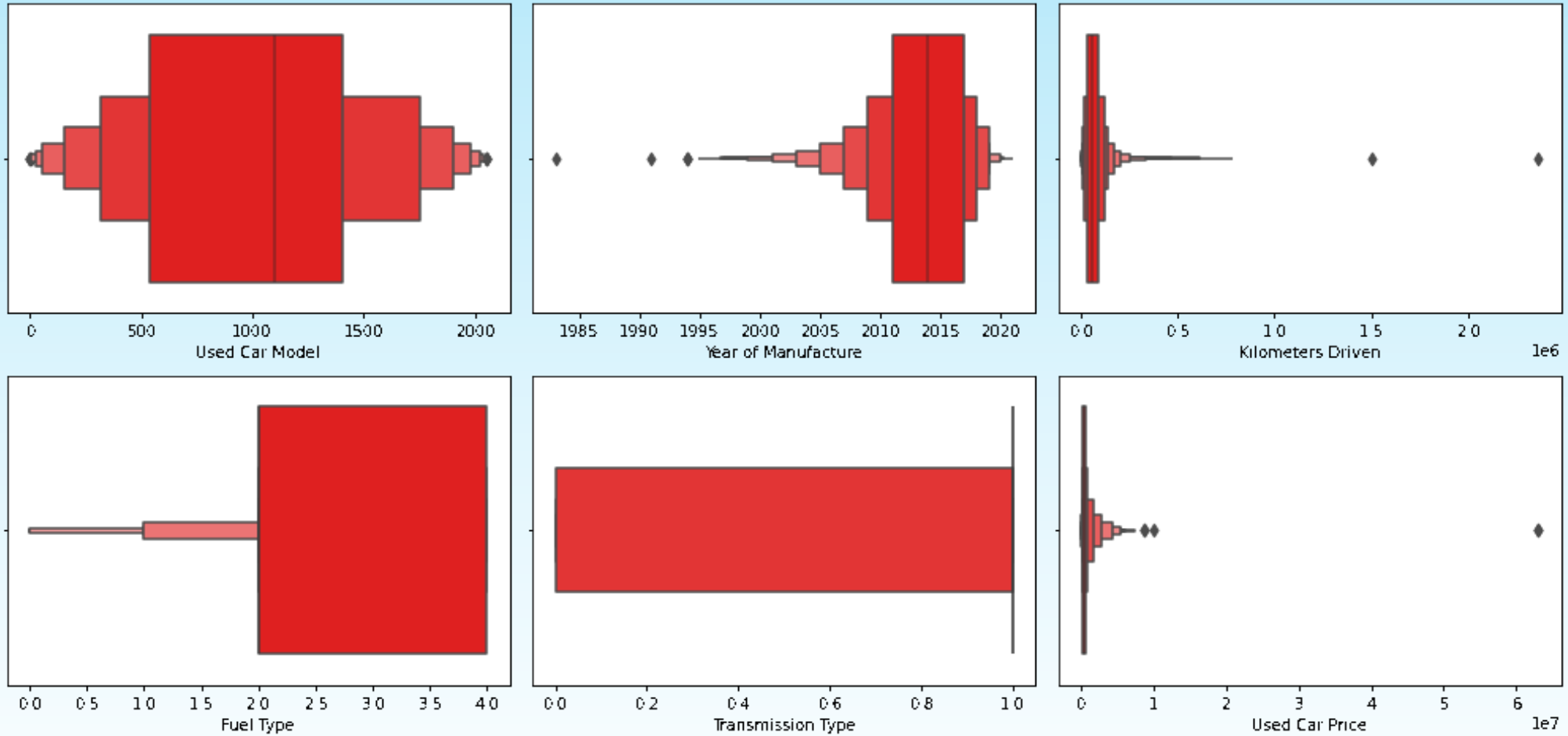
BAR PLOTS


Barplot for Year of Manufacture column vs Transmission Type column
Transmission Type
Manual
Automatic
0
250
500
750
1000
1250
1500
1750
2000
Year of Manufacture

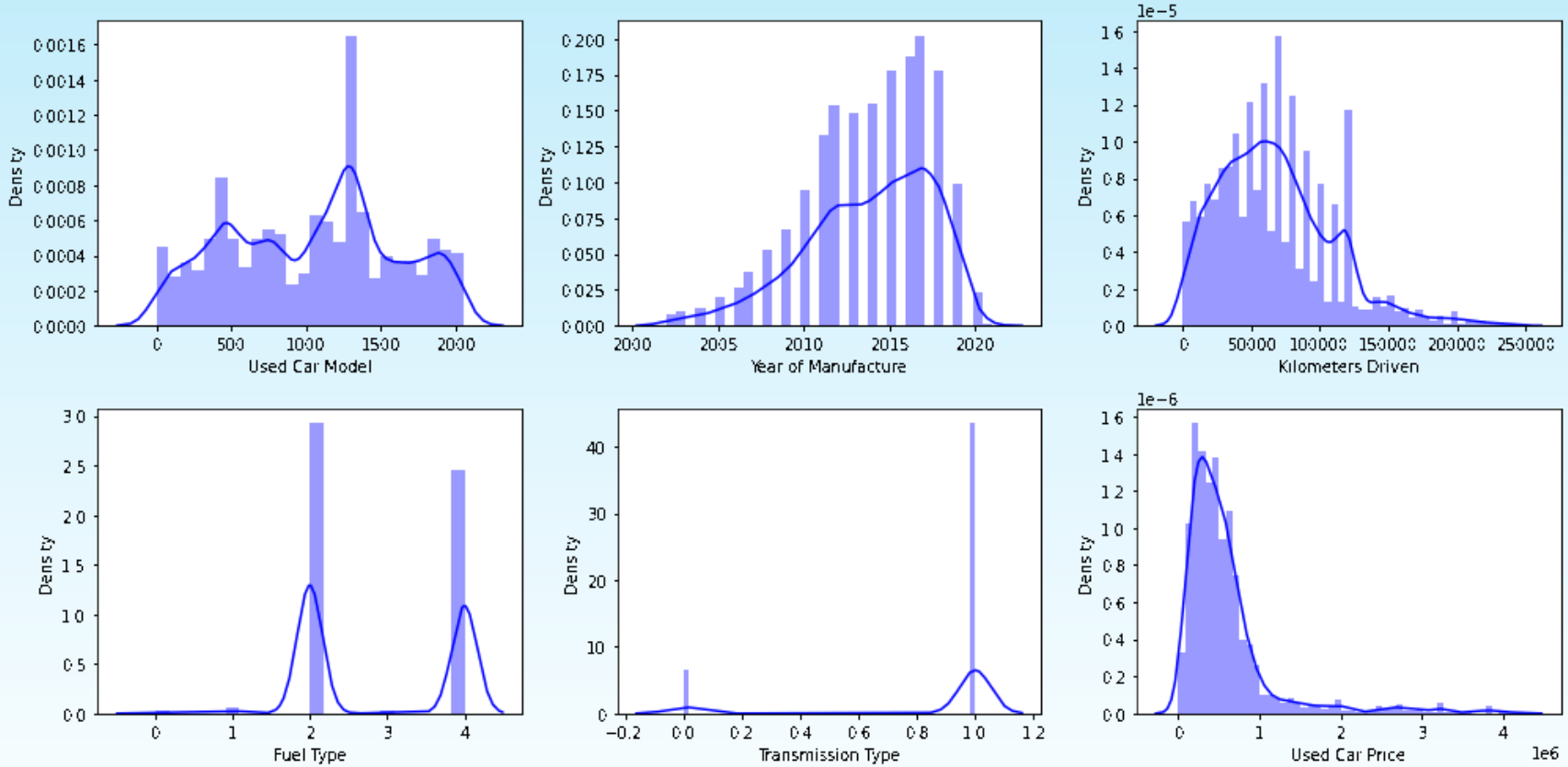
PAIR PLOTS


Year of Manufacture
Kilometers Driven
Used Car Price
1e6
1e7
Transmission Type
Manual
Automatic

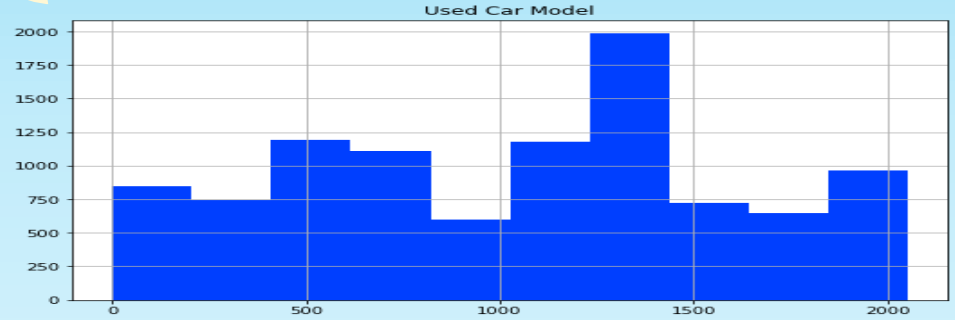
OUTLIERS WITH BOXEN PLOTS

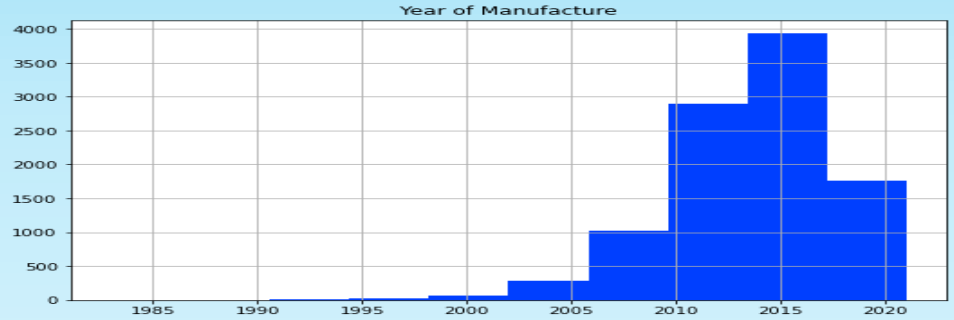


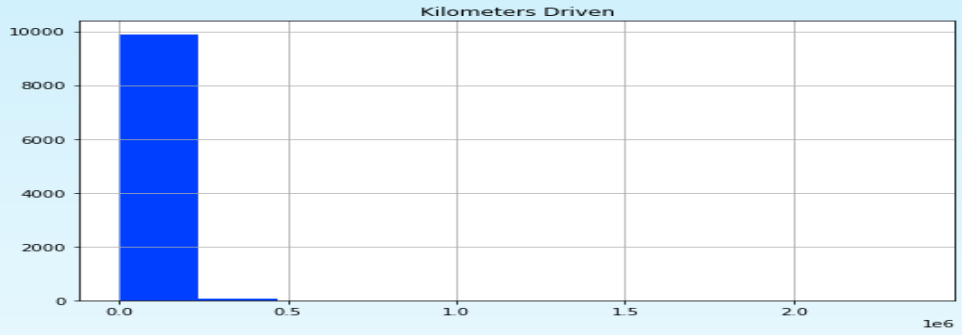
SKEWNESS WITH DISTRIBUTION PLOTS

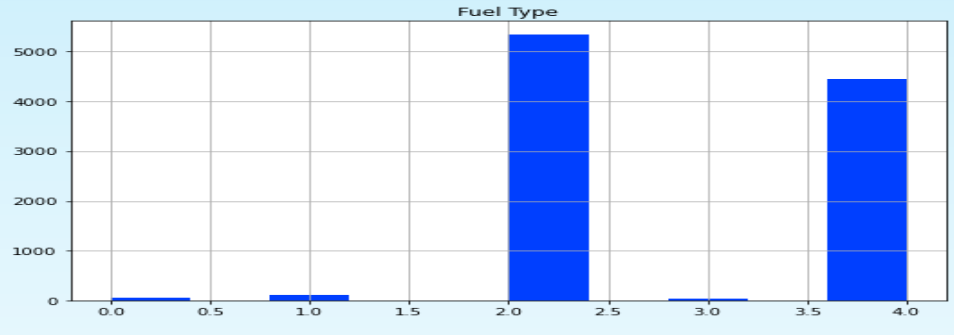


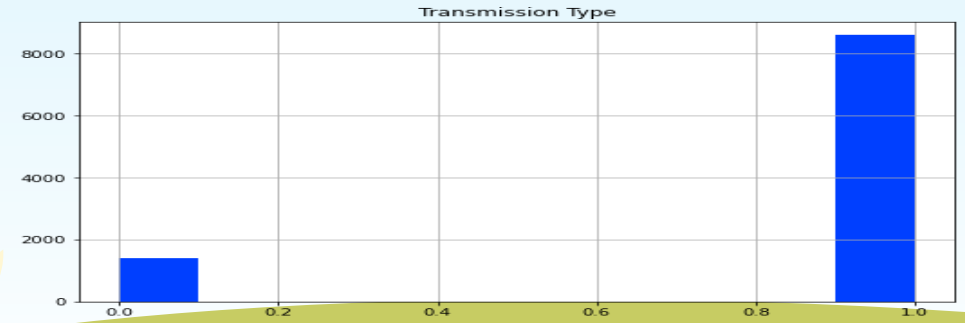
HISTOGRAM


Used Car Model
2000
1750
1500
1250
1000
750
500
250
0
0
500
1000
1500
2000


Year of Manufacture
4000
3500
3000
2500
2000
1500
1000
500
0
1985
1990
1995
2000
2005
2010
2015
2020

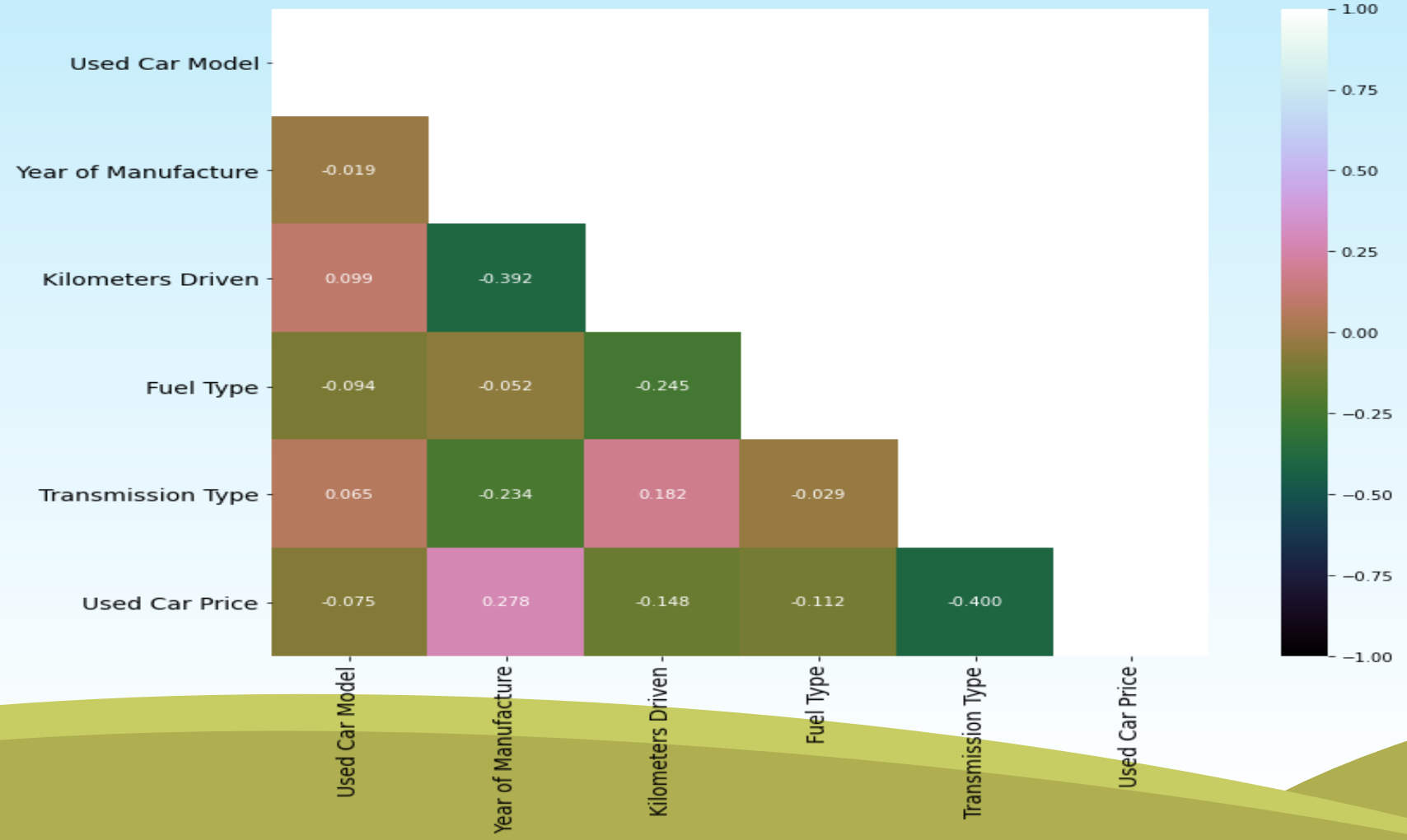

Kilometers Driven
10000
8000
6000
4000
2000
0
0.0
0.5
1.0
1.5
2.0
1e6


Fuel Type
5000
4000
3000
2000
1000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

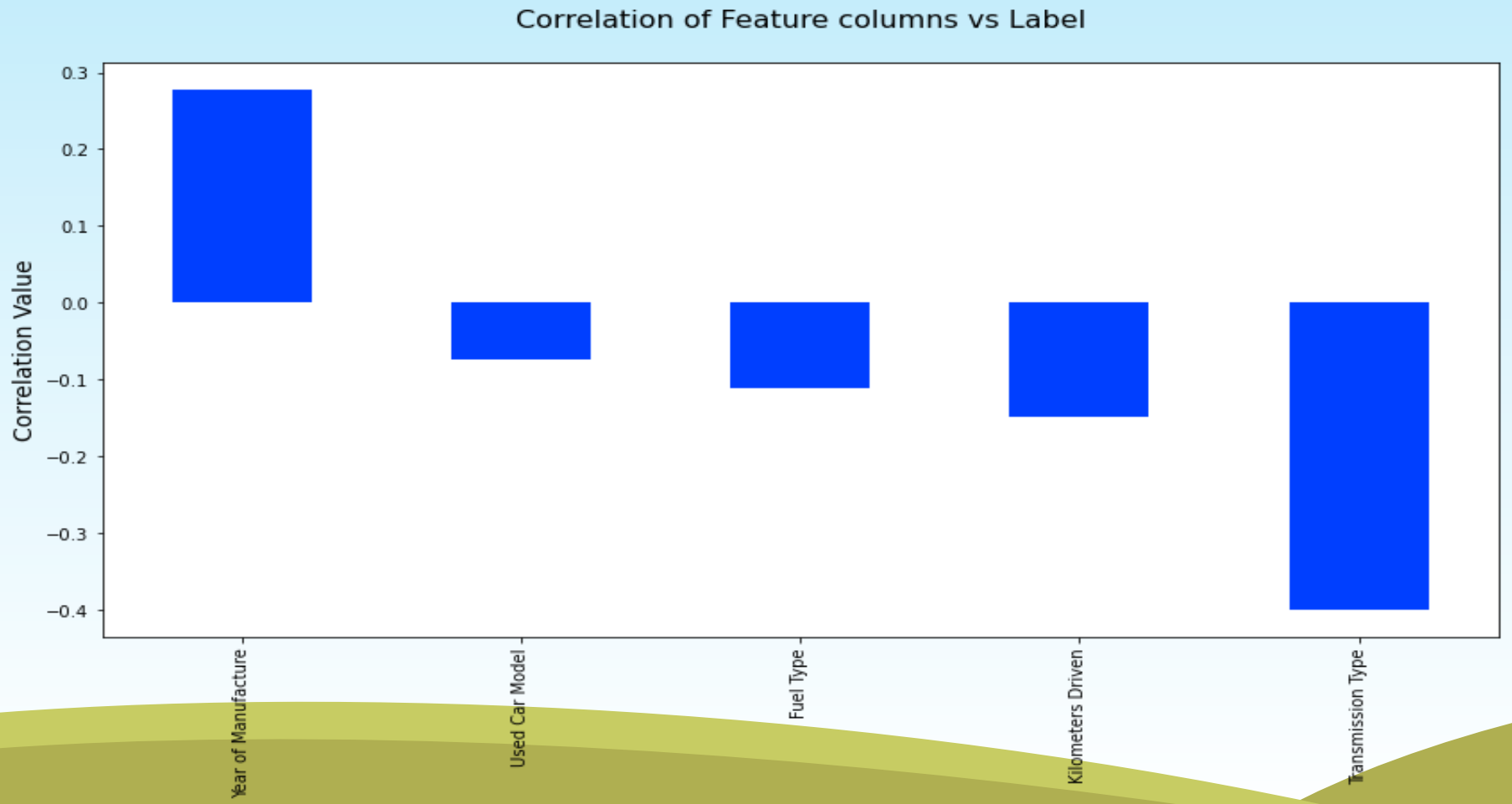

Transmission Type
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0


Used Car Price
10000
8000
6000
4000
2000
0
0
1
2
3
4
5
6
1e7

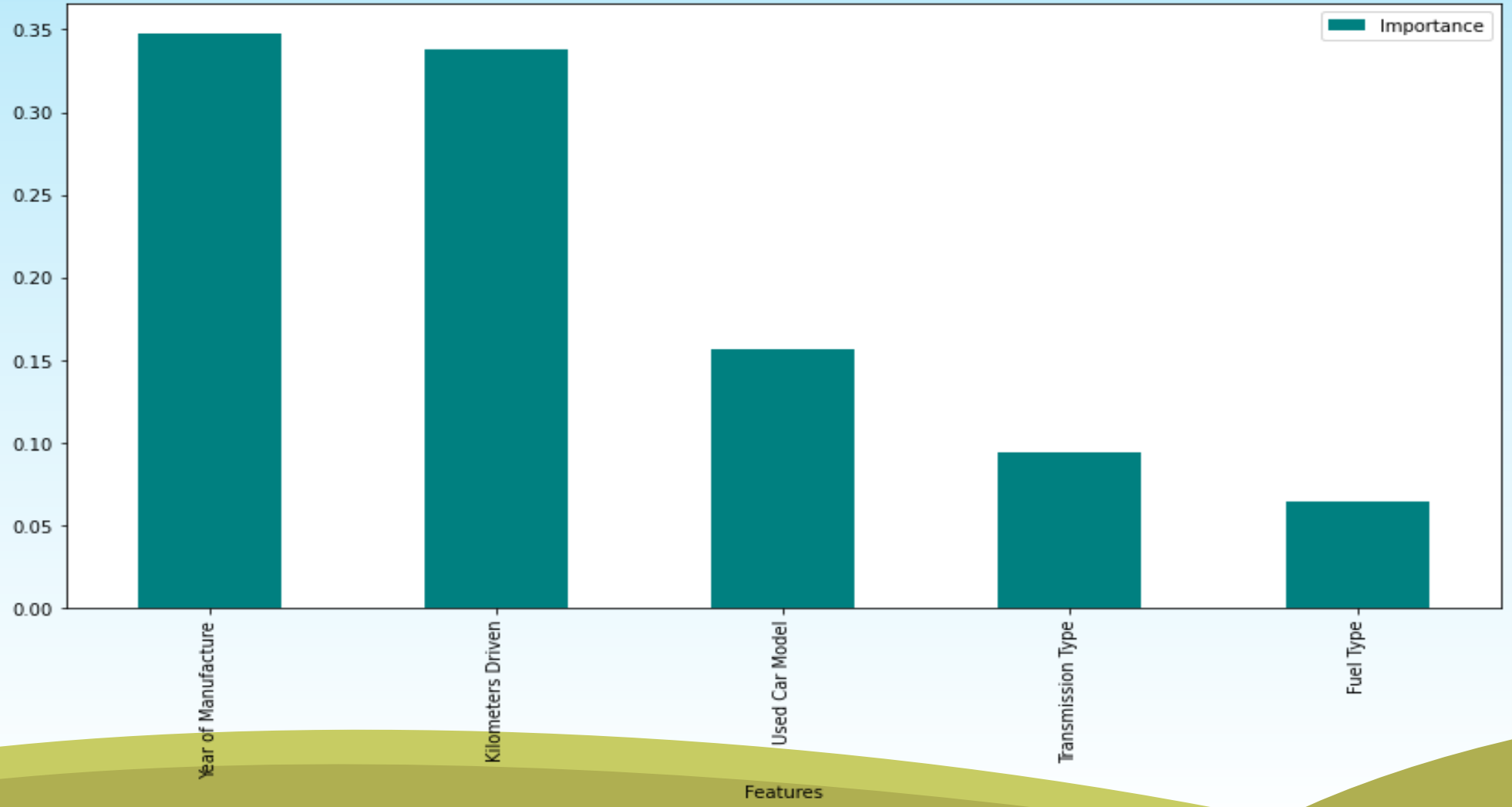
HEATMAP


Used Car Model
Year of Manufacture
Kilometers Driven
Fuel Type
Transmission Type
Used Car Price
-0.019
0.099
-0.392
-0.094
-0.052
-0.245
0.065
-0.234
0.182
-0.029
-0.075
0.278
-0.148
-0.112
-0.400
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

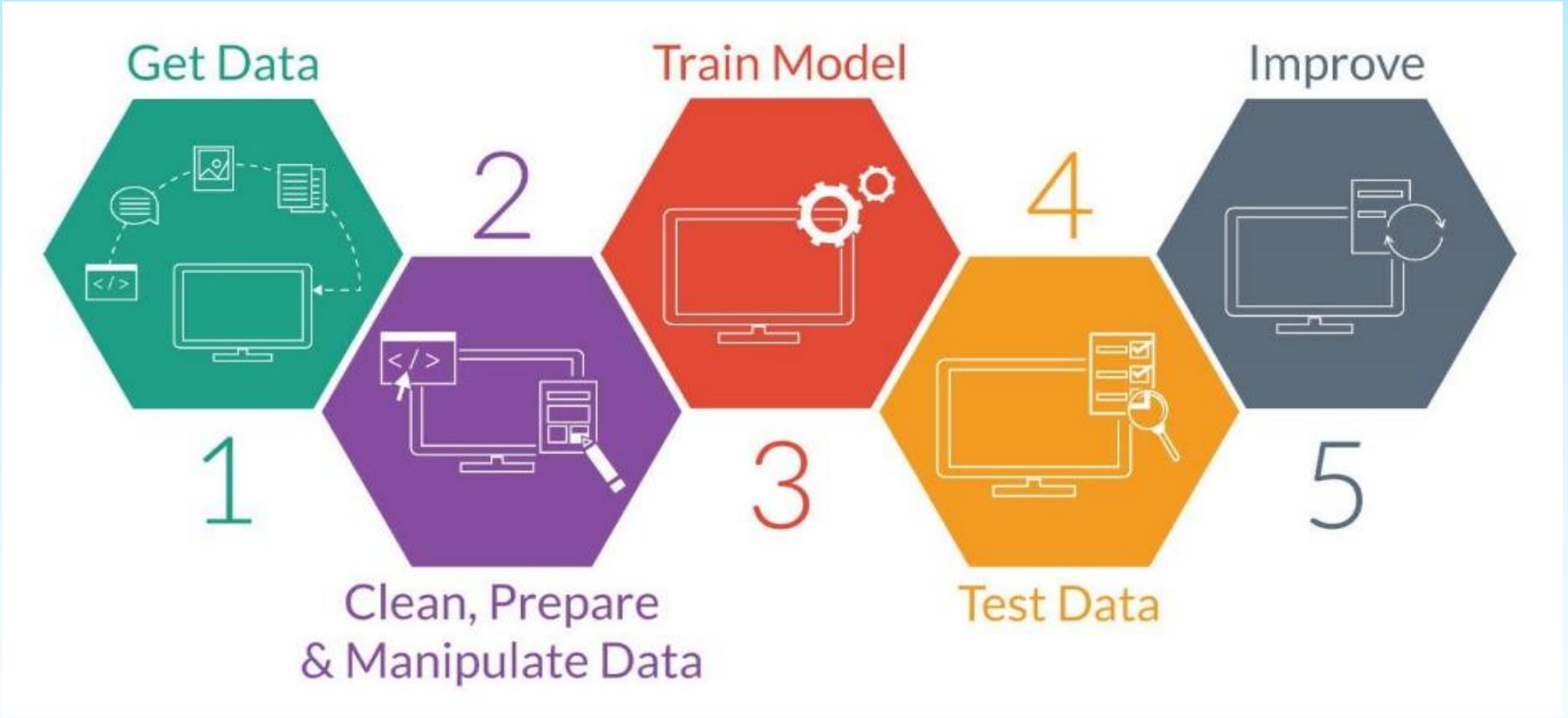
CORRELATIONS BAR GRAPH



FEATURE IMPORTANCE BAR GRAPH



MODEL TRAINING PHASES



REGRESSION MACHINE LEARNING MODEL/S USED

- Linear Regression Model
- Ridge Regularization Model
- Lasso Regularization Model
- Support Vector Regression Model
- Decision Tree Regression Model
- Random Forest Regression Model
- K Neighbours Regression Model
- Gradient Boosting Regression Model
- Ada Boost Regression Model
- Extra Trees Regression Model

REGRESSION MODEL FUNCTION WITH EVALUATION METRICS

```
# Regression Model Function

def reg(model, X, Y):
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=251)

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

    # Predicting Y_test
    pred = model.predict(X_test)

    # RMSE - a lower RMSE score is better than a higher one
    rmse = mean_squared_error(Y_test, pred, squared=False)
    print("RMSE Score is:", rmse)

    # R2 score
    r2 = r2_score(Y_test, pred, multioutput='variance_weighted')*100
    print("R2 Score is:", r2)

    # Cross Validation Score
    cv_score = (cross_val_score(model, X, Y, cv=5).mean())*100
    print("Cross Validation Score:", cv_score)

    # Result of r2 score minus cv score
    result = r2 - cv_score
    print("R2 Score - Cross Validation Score is", result)
```

RESULT OF MULTIPLE REGRESSION MODELS

```
# Linear Regression Model

model=LinearRegression()
reg(model, X, Y)
```

```
RMSE Score is: 0.6055563352393261
R2 Score is: 57.550268240713386
Cross Validation Score: 52.87533760521 49
R2 Score - Cross Validation Score is 4.674930635498484
```

EVALUATION AND HYPER PARAMETER TUNING

The key metrics used here were:

- ✓ R2 score
- ✓ Cross Validation Score
- ✓ MAE
- ✓ MSE
- ✓ RMSE

We tried to find out the best parameters list to increase our accuracy scores by using Hyperparameter Tuning. In order to achieve a higher score we used the Grid Search CV method with 5 folds.

CONCLUSION

- After the completion of this project, we got an insight on how to collect data, pre-processing the data, analyzing the data and building a model. First, we collected the used cars data from different websites like OLX, Car Dekho, Cars 24, OLA etc. and it was done by using Web Scraping.
- The framework used for web scraping was Beautiful Soup and Selenium, which has an advantage of automating our process of collecting data. We collected almost 10000 of data which contained the selling price and other related features of used cars. Then the scrapped data was combined in a single data frame and saved in a csv file so that we can open it and analyze the data.

CONCLUSION

- We did data cleaning, data pre-processing steps like finding and handling null values, removing words from numbers, converting object to int type, data visualization, handling outliers and skewness etc. After separating our train and test data, we started running different machine learning regression algorithms to find out the best performing model.

- We found that Extra Tree Regressor Algorithm was performing well according to their r2_score and cross validation scores. Then we performed Hyperparameter Tuning technique using Grid Search CV for getting the best parameters and improving the score. In that Extra Tree Regressor Algorithm did not perform quite well as previously on the defaults but we finalized that model for further predictions as it was still better than the rest. We saved the final model in pkl format using the joblib library after getting a dataframe of predicted and actual used car price details.

LIMITATIONS OF THIS WORK AND SCOPE FOR FUTURE WORK

- **The limitations we faced during this project were:**

The website was poorly designed because the scrapping took a lot of time and there were many issues in accessing to next page. Also need further practice in terms of various web scraping techniques. More negative correlated data were present than the positive correlated one's. Presence of outliers and skewness were detected and while dealing with them we had to lose a bit of valuable data. No information for handling these fast-paced websites were provided so that was consuming more time in web scraping part.

- **Future Work Scope**:

Current model is limited to used car data but this can further be improved for other sectors of automobiles by training the model accordingly. The overall score can also be improved further by training the model with more specific data.

References:

1) https://www.google.com/

2) https://www.youtube.com/

3) https://scikit-learn.org/stable/user_guide.html

4) https://github.com/

5) https://www.kaggle.com/

6) https://medium.com/

7) https://towardsdatascience.com/

8) https://www.analyticsvidhya.com/


Thank you...