FACULTY OF BUSINESS AND ECONOMICS

TITLE:

MACHINE LEARNING ANALYSIS ON USED CAR PRICES PREDICTION

REVIEW BY:

EUGENE ADU-GYAMFI

ROSHAH CHARISMA S. NOMO

Introduction

Machine learning generates a usable model or program by evaluating a large number of solutions against the given data and selecting the one that best fits the situation. As a result, machine learning can be useful for solving problems that require much human effort. It can efficiently and accurately inform judgments and make predictions about complex topics. It also aids with data analysis, expanding to predictions for accurate decision-making. Recent advances in machine learning make it possible to design efficient prediction algorithms for data sets with huge numbers of parameters (Gammerman & Vovk, 2007). This notion extends to the automobile sector, particularly, purchasing used cars. Various factors such as mileage, model, year of the car purchased, vehicle identification number, manufacturer of the car, city, and state it was sold which affect the prediction of prices.

Objective

This study aims to predict the pricing of used cars based on a range of features.

Methodology

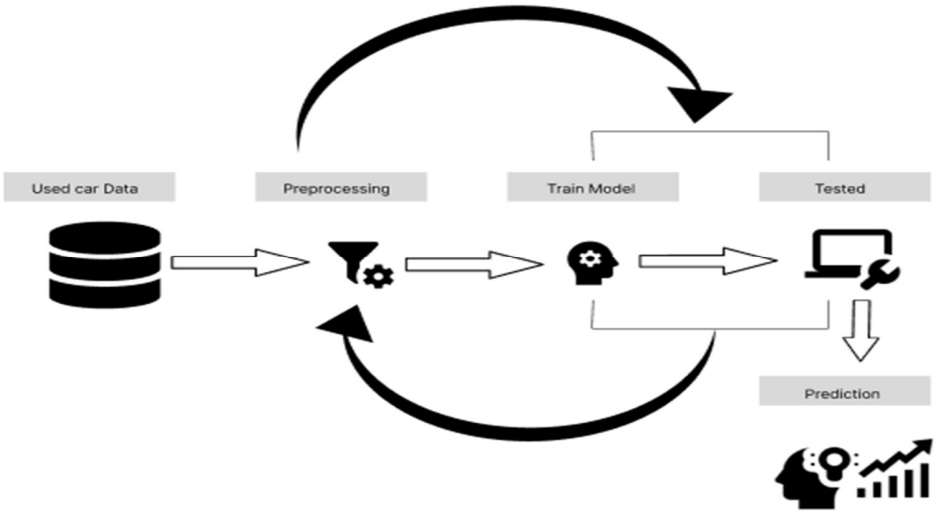

Fig 1. Proposed Methodology

Data Description

The data is sourced from https://www.kaggle.com/datasets/harikrishnareddyb/used-car-price-predictions. It contains 852,122 data samples with 8 features comprising seven independent variables (that is: mileage, model, year of the car purchased, vehicle identification number, manufacturer of the car, city and state it was sold) and one dependent or target variable (Price). The independent variables are beneficial for developing predictions for predictive modeling in this project.

Installing and Importing Dependencies, Data

The data as previously mentioned was downloaded from Kaggle and uploaded to Colab via Google Drive. The various dependencies or libraries included but not limited to Pandas, Numpy, Seaborn, Matplotlib, Scikit amongst others.

Exploratory Data Analysis

In this phase, the data was explored to gain more understanding or depth into it. We followed the process of visualizing, exploring the statistical features of the data involved. The Pandas library was used to explore the data, its type as well as statistical features and the seaborn used to visualize it to decide the direction to pursue.

```
[ ] #Viewing data
    df.head(10)
```

	Price	Year	Mileage	City	State	Vin	Make	Model
0	8995	2014	35725	El Paso	TX	19VDE2E53EE000083	Acura	ILX6-Speed
1	10888	2013	19606	Long Island City	NY	19VDE1F52DE012636	Acura	ILX5-Speed
2	8995	2013	48851	El Paso	TX	19VDE2E52DE000025	Acura	ILX6-Speed
3	10999	2014	39922	Windsor	CO	19VDE1F71EE003817	Acura	ILX5-Speed
4	14799	2016	22142	Lindon	UT	19UDE2F32GA001284	Acura	ILXAutomatic
5	7989	2012	105246	Miami	FL	JH4CU2F83CC019895	Acura	TSXAutomatic
6	14490	2014	34032	Greatneck	NY	JH4CU2F84EC002686	Acura	TSXSpecial
7	13995	2013	32384	West Jordan	UT	JH4CU2F64DC006203	Acura	TSX5-Speed
8	10495	2013	57596	Waterbury	CT	19VDE2E50DE000234	Acura	ILX6-Speed
9	9995	2013	63887	El Paso	TX	19VDE1F50DE010450	Acura	ILX5-Speed

Fig 2. Tabular View of Data

```
#Exploring the datatype
df.dtypes
```

```
Price       int64
Year        int64
Mileage     int64
City       object
State      object
Vin        object
Make       object
Model      object
dtype: object
```

Fig 2.1 Datatype of Variables

```
#Exploring the statistics of all the numerical value in the data
df.describe()
```

	Price	Year	Mileage
count	852122.000000	852122.000000	8.521220e+05
mean	21464.100210	2013.289145	5.250779e+04
std	13596.202241	3.414987	4.198896e+04
min	1500.000000	1997.000000	5.000000e+00
25%	13000.000000	2012.000000	2.383600e+04
50%	18500.000000	2014.000000	4.025600e+04
75%	26995.000000	2016.000000	7.218600e+04
max	499500.000000	2018.000000	2.856196e+06

Fig 2.2 Descriptive Statistics of Data

The data as also checked for any missing values in the attributes, but none was found as indicated below:

```
#Checking for missing values in the data
#missing values can be filled intelligently wi
df.isnull().sum()
```

```
Price      0
Year       0
Mileage    0
City       0
State      0
Vin        0
Make       0
Model      0
dtype: int64
```

Fig 2.3 Missing values

In gaining a further understanding into the data, a pairplot was done for visualization to check for possible outliers and identify the right way to approach the pre-processing. Also, the various quantitative variables were visualized as against the target variable.

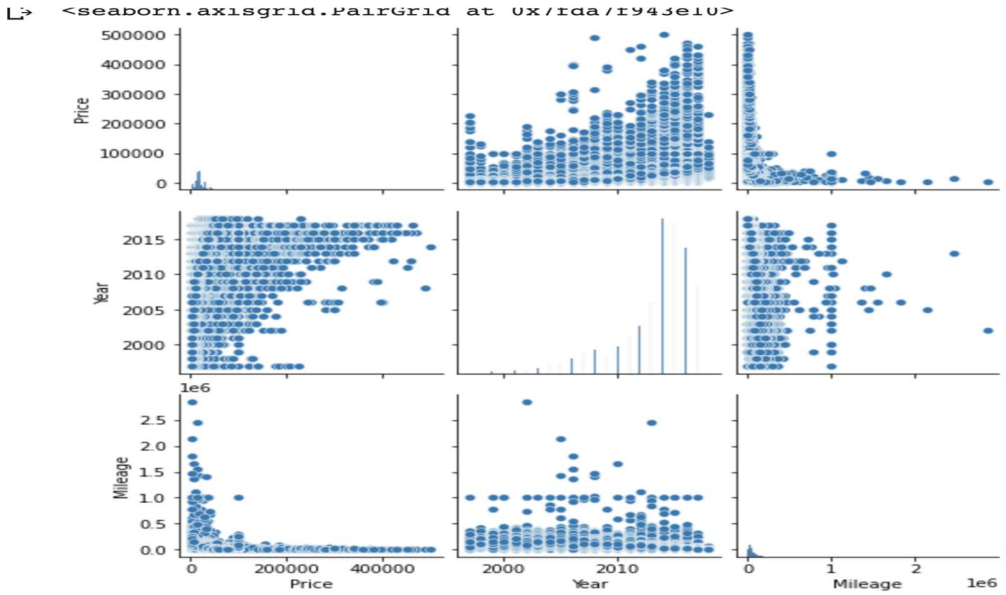

Fig 2.4 Pair plot visualization

PREPROCESSING

It is the process of making raw data more suitable for machine learning models. As highlighted in our machine learning class, that a well preprocessed data often improves results better than tuning the algorithm, we analyzed the best possible way to go about it.

A boxplot of the target variable showed us outliers which we used the following commands to correct the data as best as we could without not losing much to influence the outcome negatively.

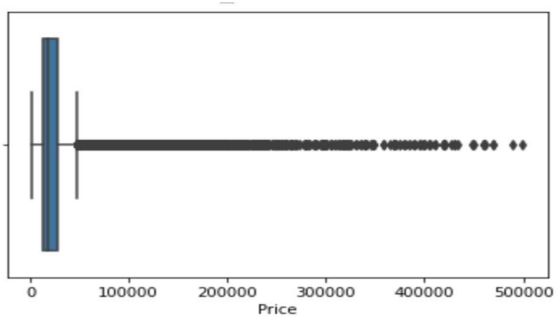

Fig 3 Boxplot

```
[18] #Removing some of outliers
     df2 =df[df.Price <=46000]
```

Fig 3.1 Removing outliers

```
[19] #Visualizing the data after removing outliers
     sns.boxplot(df2.Price)

     /usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable
       FutureWarning
     <matplotlib.axes._subplots.AxesSubplot at 0x7fda72582fd0>
```

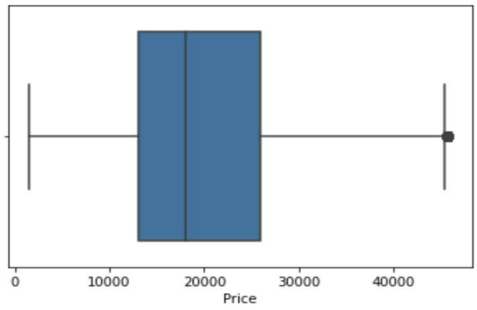

Fig 3.2 Data visualization after removal of outliers

The histogram plot also provided guidelines in removing the outliers as not much core data was removed and the skewness of the data to the right was also corrected slightly.

For the second phase of the model building we added more features to the data for training and testing to improve upon the accuracy. We encoded the features with string data type and added it for the training. However, we were cautious as to not include more than necessary features to prevent overfitting that can negatively affect the testing and predominantly application of the model in real life related data.

FEATURE SELECTION

This is where useful features (variables) are selected in and used to build the model. In the instance where redundant variables are added, it can affect the accuracy of the classifier by reducing its generalization capability. On the other hand, more variables can increase the overall complexity of the model. (Brownlee, 2020). The reason why we used feature correlation was to identify relevant variables to help with the prediction as highly correlated variables indicate that it can be used to predict one from the other.

The correlation plot below indicates that price and year are the features that have strong correlation and are relevant to include in the training.

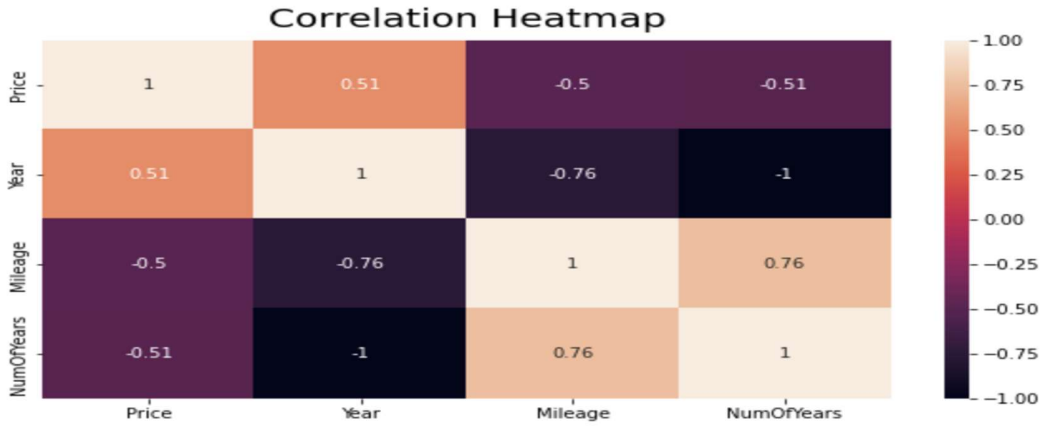

Fig 4. Correlation Heatmap

DATA SPLITTING

For the training of the ML algorithms, we used the conventional 20% and 80% split of the data for the testing and training accordingly.

DATA TRAINING, TESTING & EVALUATION

ALGORITHMS USED

The first algorithms explored for our model building was the linear regression and K-Nearest Neighbor regression as the dependent and independent variables are both quantitative. The Linear Regression Model helps in establishing a relationship between the variables in question so prediction can be made as shown in the figure below:

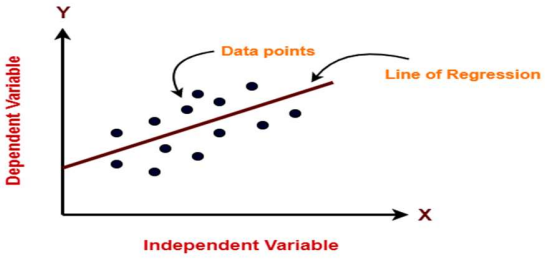

Fig 5. Linear Regression

On the other hand, the KNN algorithm can be used for both classification and regression, so we explored the regression part in on the data. It normally works by identifying the similarity between features to predict values of any new data points given.

After first Preprocessing

	Training Score	Testing Score	Cross_Validation Score	r2_Score	Mean Absolute Error	Mean Squared Error
Linear Regression	27.58%	27.56%	27.58%	27.56%	6367.34	7864.98
K-nearest neighbor regression	44.17%	17.33%	16.87%	17.33%	6681.30	8401.95

After Second Preprocessing (Adding more features)

	Training Score	Testing Score	Cross_Validation Score	r2_Score	Mean Absolute Error	Root Mean Squared Error
Linear Regression	27.58%	27.56%	27.57%	27.56%	6367.34	7864.98
K-nearest neighbor regression	88.40%	86.05%	85.80%	86.05%	2440.97	3450.96

Another approach we considered was to use a classification algorithms. In doing that we set the target variable which was numerical to categorical value, “low” or “high” by using a threshold around the value of the mean price. Also making sure the lows and high values counts are set closely at par.

Changing the numerical value of Price to categorical value

```
[248] df2["Price_bins"]=df2.Price.apply(lambda x: "high" if x> 18500 else "low" )
```

Fig 6. Numerical Value to Categorical Value

```
df2.Price_bins.value_counts()

low     426991
high    392543
Name: Price_bins, dtype: int64
```

Fig 6.1.

The following models were considered as our data falls under supervised machine learning. With the two algorithms (logistic regression and Support Vector Machine) using a linearly separable approach for its classification. The K-Nearest Neighbor Classifier uses the same approach as the one for regression.

After first Preprocessing

	Training Score	Testing Score	Cross_Validation Score
Logistic Regression	66.68%	66.75%	66.68%
K-Nearest Neighbor Classifier	75.28%	63.55%	63.42%

Confusion Matrix

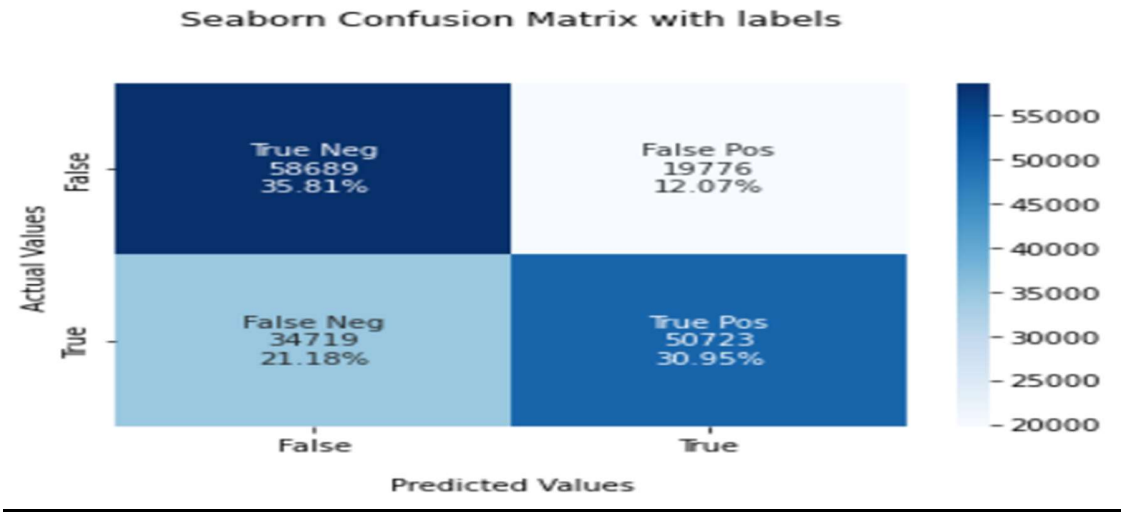

Fig 7. Logistic Regression-Confusion matrix

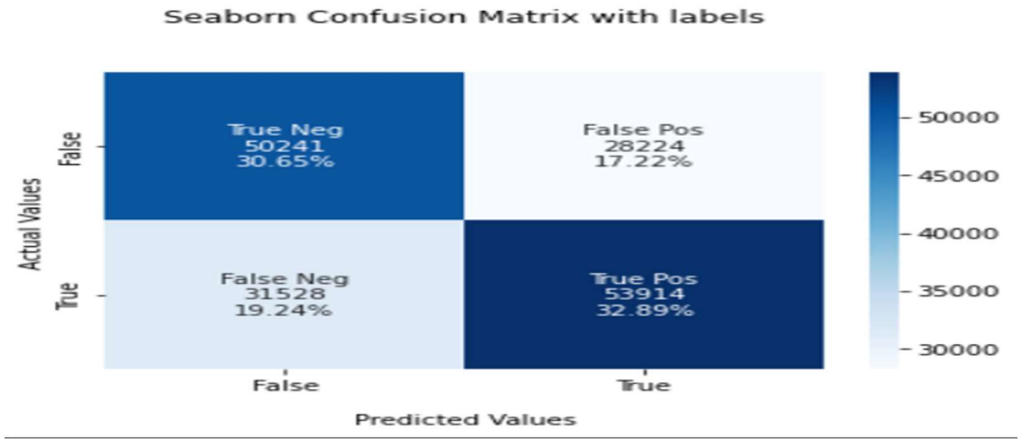

Fig 7.1 KNN-Confusion Matrix

After Second Preprocessing

	Training Score	Testing Score	Cross_Validation Score
Logistic Regression	54.03%	53.95%	54.03%
K-Nearest Neighbor Classifier	91.17%	89.64%	89.70%
Support Vector Machine	66.47%	66.67%	66.64%

Confusion Matrix

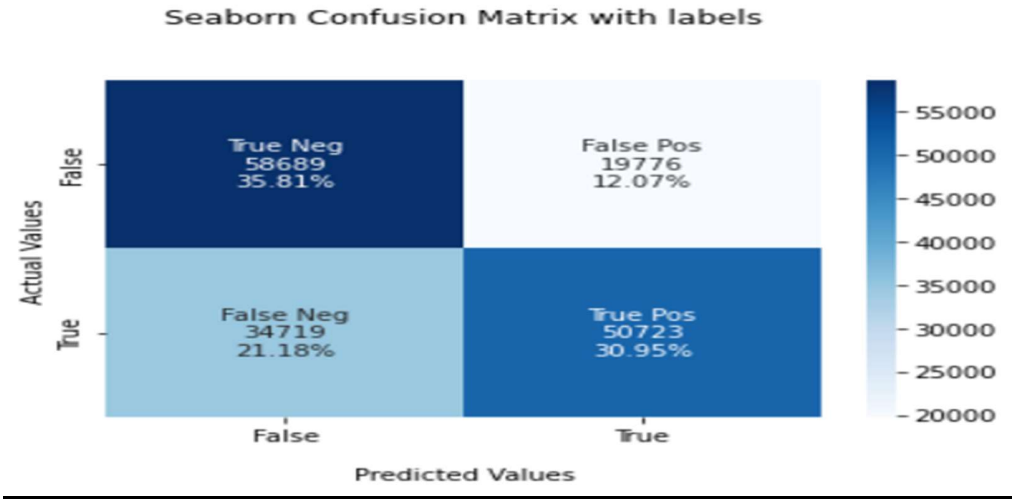

Fig 8. Logistic Regression-Confusion Matrix

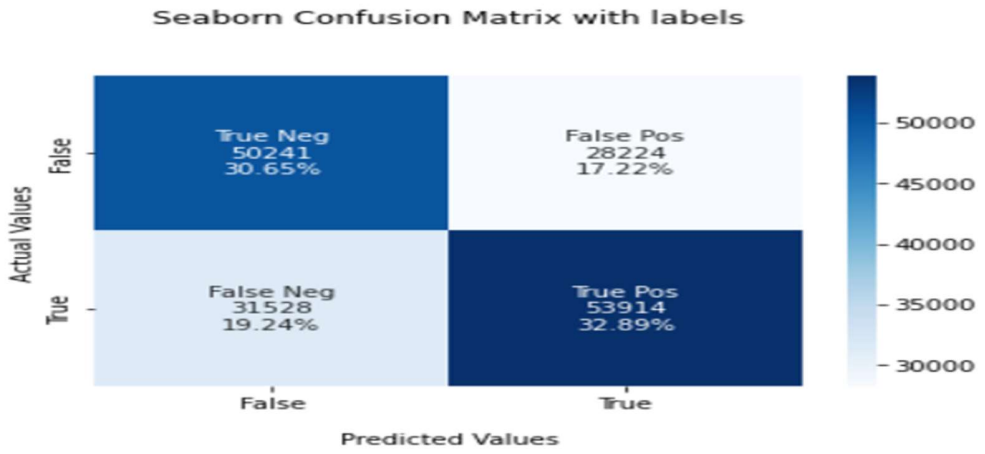

Fig 8.1. KNN- Confusion Matrix

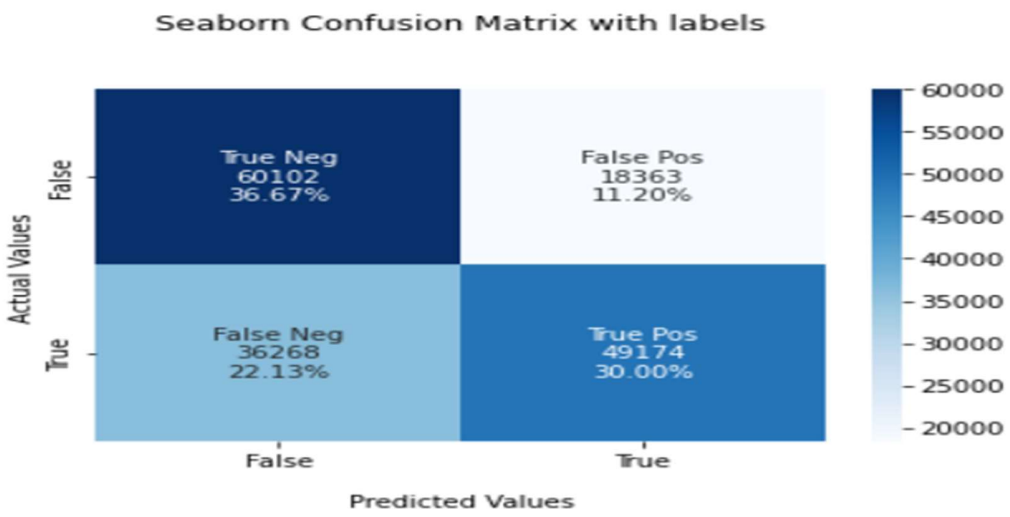

Fig.8.2 Support Vector Machine -Confusion Matrix

CONCLUSION

Following an investigation of the methods utilized, the K-Nearest Neighbor Regression algorithm assisted with the best fit line and could more accurately predict the price. We opted for that one because the Regression algorithms are used to predict continuous values like price, salary etc whereas classification algorithms are used to predict discrete values such as true or false, yes or no and in our case whether the price was low or high.

THINGS WE LEARNT

1. It is important to visualize the data to gain insight
2. It is relevant to identify the type of data you have in order to know the approach (supervised or unsupervised learning) to use.
3. Adding more or a reasonable number of features can help improve the training of the algorithm.
4. Label encoding can be utilized in converting other data types to numeric form which are readable for the machine.
5. It is important to use a ML algorithm suitable for the model building

Python Code link

https://colab.research.google.com/drive/1IascXW_NFv1bZ8gvODoR96m89VQK89bh?usp=sharing

REFERENCES

A. Gammerman and V. Vovk, "Hedging Predictions in Machine Learning," in *The Computer Journal*, vol. 50, no. 2, pp. 151-163, March 2007, doi: 10.1093/comjnl/bxl065.

Asiri, S. (n.d.). *Machine Learning Classifiers*. Towards Data Science. Retrieved May 23, 2022, from https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

Jason Brownlee. (2020). *How to Choose a Feature Selection Method For Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

Regression vs Classification in Machine Learning - Javatpoint. (2022). Retrieved 29 May 2022, from https://www.javatpoint.com/regression-vs-classification-in-machine-learning