**INFO6105 - Data Science Engineering Methods**

**Project Report – Craigslist Used Car Price Prediction**

**Project Group 6:**

1. Neeraja Dixit – 001584742
2. Kunal Desai – 002193570

**INTRODUCTION:**

* **Background:**

In the past few years, price of the used cars is escalating. The manufacturer sets the price of new cars in the industry, with the government incurring some additional costs in the form of taxes. As a result, buyers purchasing a new car may be confident that the money they invest will be well spent. However, due to the higher price of new automobiles and customers' inability to purchase new cars due to a lack of cash, used car sales are on the rise globally.

Craigslist has the most used cars for sale in the world. However, finding them all at the same time is extremely difficult. This dataset aids in the collection of aggregate data for all used cars in the United States.

This dataset will be used to forecast the value of used automobiles. It includes information about a car as well as the price at which it was sold. This dataset is critical because it contains sales data from the previous year and will be used to train the machine learning model.

The price of an automobile will be approximated based on the product's many qualities. Traditionally, the price of secondhand cars was forecast based on the subjective opinions of experts. However, when utilizing machine learning algorithms to anticipate the price of a used car, we took into account a number of parameters at the same time. It is difficult for a human individual to absorb this much data and determine the optimum price for a used automobile on the market.

* **Motivation:**

The motivation behind this initiative is to assist buyers and sellers in their transactions in acquiring a better grasp of the pricing of a used car.

Not everyone will be familiar with the estimated price of a used car. When forecasting the price of a used car, many factors should be examined, including the region, entry price, model, year, fuel type, and car condition. Obtaining the best results for a used car's estimated worth can assist sellers in maximizing revenues while also assisting purchasers in obtaining the best price.

Given the numerous elements that influence the pricing of a car and the massive quantity of data that will be needed to train the model, it is extremely difficult for a human to process this amount of data and make correct predictions. In this situation, the model assists by making it easier for less experienced car owners/buyers, as well as experienced people, to make a transaction with a higher possibility of receiving accurate results.

* **Goal:**

The goal of this project is to create a model for estimating the accurate value of a used car. Since 2022 due to the pandemic, used car prices have increased significantly. Such unusual circumstances affect the market and the people who have to make the decision of either buying or selling their cars.

Different factors, such as the car's condition and size, have a part in predicting the price of an automobile. The price of an automobile may also vary depending on the year and market trends in that year. The price of a car is impacted by a number of factors, including its type, transmission, and drive type. When developing a model, we took into account all of these factors, which resulted in a more accurate outcome.

We assessed the model's accuracy when it was finished to see how it compared to real-world automobile costs and how well it performed. A high-accuracy model will help both buyers and sellers maximize transaction revenues.

**METHODOLOGY:**

* **Importing required libraries:-**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import plotly.express as px

import warnings

warnings.filterwarnings('ignore')

from sklearn.preprocessing import OrdinalEncoder

from sklearn import preprocessing

from tqdm import tqdm

from sklearn.experimental import enable\_iterative\_imputer

from sklearn.impute import IterativeImputer

from sklearn.linear\_model import BayesianRidge

from sklearn.preprocessing import StandardScaler

from sklearn.preprocessing import MinMaxScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_log\_error,r2\_score,mean\_squared\_error

import matplotlib

import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestRegressor

import xgboost as xgb

* **Data Preprocessing/Wrangling/Cleaning:**

The dataset we're working with was scraped from the Craigslist website. As a result, some characteristics are meaningless for price prediction. The dataset contains many attributes which might only go towards making the modelling and prediction process more inaccurate. Such features/attributes is better to be removed from the dataset using proper methods and then apply modelling techniques on them. These features will be removed from the dataset.

Below are the features which will be removed from the dataset which doesn’t make any sense:-

* 'Unnamed : 0'
* 'id'
* 'url'
* 'region\_url'
* 'image\_url'
* 'VIN'

On further checks, we found that the attribute ‘description’ as well doesn’t add any value towards the prediction of price. Hence, we have dropped that column as well from the dataset. Post this we were left with a dataset of below shape:-

* **(Rows, Columns):- (458213, 18)**

Post this, we checked for null/missing values in each feature. We created a function to check the same. This function was able to iterate through each column and find the total percentage of missing values in the column. When we ran this function we found that the Size column had missing values of about 70%. Realistically thinking, any column containing such huge amount of NULL values would only prove to bring the prediction algorithm bad results. We also checked for NULL values using the Heatmap from Seaborn library, which again showed the containment of huge amount of NULL values in Size attribute. This led us to the conclusion that dropping the Size column as well was logical. Hence, we have dropped that column as well from the dataset.

Now, from the remaining number of columns, we had to handle the NULL values. To handle or impute the NULL values from Numerical features we used the Mean of the columns and to impute the NULL values in the Categorical features, we used the Label Encoding method.

Heatmap showing NULL values in the dataset. Huge amount of NULL values present in Size column being highlighted below.

Chart, bar chart

Description automatically generated

With this we were done with handling all the missing values in our dataset. We then turned to our next milestone of checking the amount of outliers in our numerical features. The numerical features like Price, Odometer, Year contained fair amount of outliers which were plotted using box plot and then removed using a systematic method.   
 It is always good to know that outliers can lead to rendering a bad prediction model. Hence, its extremely essential that we remove all the outliers from the dataset and then pass it further for modelling.

We'll deal with outliers in the data after dealing with null values. Outliers are removed from the data using the IQR (InterQuartile Range) approach. The IQR method is the best way to handle and deal with outliers in the dataset.

Below screenshots help in knowing what amount of outliers were present in which columns. Further, how were they handled in the dataset.

Outliers are prices with values less than 6.28 and more than 12.19.

Chart, diagram

Description automatically generated

Here, the outlier range is not visible but we have it stored in the notebook which deletes the appropriate extra records.

Chart

Description automatically generated

Outliers include years prior to 1995 and after 2020.

Chart

Description automatically generated

* **Data Visualization:**

This is the section where we received huge amount of insights from the dataset. Here we understood the correlation between columns. How different attributes in the dataset are related to the target variable. What is the different count of attributes in the dataset. How does price vary in the database and also a pair plot which helped us with huge insights about all the numerical data relations.

Below are some data visualizations that we have embedded in the notebook which gives us a fair bit of idea about how attributes in the dataset relates to each other.

Below Map showing the distribution and variance of Prices of cars in all the states in Unites States.

Map

Description automatically generated

Below is the visualization which helps us run through the different types of transmission types given the year in the X axis.

Graphical user interface, application

Description automatically generated

Below bar graph shows the relationship between the condition of the car and the amount of variance Price has in regards with it.

Chart, bar chart

Description automatically generated

Below bar graph shows the relationship between the number of Cylinders the car has and the amount of variance Price has in regards with it.

Chart, bar chart

Description automatically generated

Below bar graph shows the count of different cars based on its condition.

Chart, bar chart

Description automatically generated

Below bar graph shows the count of cars, year-wise. That is, in which year, what specific amount of cars were put on sale.

Chart, bar chart, histogram

Description automatically generated

Below bar graph shows the count of car’s different transmission type.

Chart, bar chart, waterfall chart

Description automatically generated

Below is the sample of Pair Plot that was created to show the relationship between all the numerical attributes that we have in the dataset.

Chart, histogram, scatter chart

Description automatically generated

Below Distribution graph shows the skewness and distribution of the Price in the dataset. Here we can see that the price in the dataset is right skewed.

Chart, histogram

Description automatically generated with medium confidence

* **Label Encoding:**

There are 12 features in our dataset that contain values in the form of labels. These labels need to be changed to a machine-readable format. As a result, we must convert them into numerical variables. LabelEncoder from the Sklearn library is used for this. These categorical values when converted into numerical values using Label Encoding, we get a dataset which can be forwarded to the machine learning models for accurate prediction of the target variables.

* **Data Scaling/Normalization/Standardization:**

The data is not dispersed normally. The data must be processed in order for the columns to be transformed to the same scale. As a result, these two traits have radically distinct ranges.

When we evaluate the data further, the attributed income will have a greater influence on the result if a particular feature has a higher value. This does not, however, imply that it is a more accurate prediction. As a result, variables which contain values where there is a huge difference in range are normalized.

To normalize the data, we use the sklearn module MinMaxScaler.

* **Machine Learning Modelling:**
* **Linear Regression Algorithm:**

One of the modeling algorithms we use for this dataset would be Linear Regression. Linear Regression is a Supervised learning model. Here the algorithm tries to predict the value of the dependent variable by finding a best-fit regression line between the independent and the dependent variables.

Linear regression can be used between two or multiple variables. In the Linear Regression algorithm, the relationships are represented using linear predictor functions whose unknown model parameters are derived from the data. Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), consequently called linear regression.

The sort of link between a predictor variable and the response variable is shown by the coefficients. When the independent or predictor variable rises, the dependent or response variable also increases, resulting in a positive sign. A negative sign denotes the inverse. When the value of an independent variable rises, the response variable falls.

The features with positive coefficients are chosen as the most important independent or predictive variables for further analysis and plotted on a graph.

Result of linear regression algorithm:

Chart

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

* **Random Forest Algorithm:**

We will also use the Random Forest Regression technique, which is a supervised learning model once again. Random Forest is a model for ensemble learning. The ensemble modeling technique integrates several classifiers to solve a complex problem. Having used multiple classifiers increases the model's performance hugely. For regression tasks, the mean or average prediction of the individual trees is returned. Deep decision trees may suffer from overfitting, but random forest avoids that by growing trees on random subsets. As a result, it's a good model to use in the analysis.

In our model, we are creating 180 decisions with max\_features 0.5.

To find out variable importance, we have plotted a graph to find out relative importance of all the variables. A bar plot is constructed for all of the attributes in order to identify the most important ones. The bar graph demonstrates that a car's year is the most crucial feature in predicting its sale value. The odometer is the most crucial feature to estimate a car's value after the year, followed by the cylinders, fuel, and model.

Result of Random Forest algorithm:

**Chart, bar chart, histogram

Description automatically generated**

**Chart, histogram

Description automatically generated**

* **XGBoost Algorithm:**

XGBoost is a machine learning modeling technique where gradient boosted trees algorithm is implemented. XGBoost is one of the popular and efficient open-source implementations. Gradient boosting is a supervised learning approach that combines the estimates of a set of smaller, weaker models to attempt to accurately predict a target variable. XGBoost trains the model very fast and it can be parallelized by distributing the computing across clusters. It uses both the software and hardware optimization techniques in order to increase the performance while maintaining the accuracy of the result.

In this model of XGBoost, 200 decision trees are created of 24 max depths. This model is learning the parameter with a 0.4 learning rate.

To find out the most important feature of a car, a bar plot is created. According to XGBoost, the odometer feature is most important to find out the price of a car. After that, the geographic location is given importance with latitude and longitude being in the following position.

Result of XGBoost algorithm:

Table

Description automatically generated

**Final set of results comparing all the Models and their accuracy and other details of the models.**

Table

Description automatically generated

**DESCRIPTION OF DATASET:**

1. **id** :- This column simply contains the unique entry id.
2. **url** :- Link containing the listing of the vehicle.
3. **Region** :- Region of the listing of the vehicle.
4. **Region\_url** :- URL link to the listing location of the vehicle
5. **Price** :- The dependent variable containing the price of the vehicle to be predicted.
6. **Year** :- Manufacturing year of the vehicle
7. **Manufacturer** :- Manufacturer of the vehicle.
8. **Model** :- Column specifying the Model of the vehicle
9. **Condition** :- This column specifies the condition of the vehicle
10. **Cylinders** :- Variable to specify the number of cylinders in the vehicle
11. **Fuel** :- The type of fuel being used by the vehicle
12. **Odometer** :- Column specifying the odometer reading i.e. the miles travelled by the vehicle.
13. **Title\_status** :- Column specifying the title status of the vehicle.
14. **Transmission** :- Column to specify the mode of transmission of the vehicle.
15. **VIN** :- This column specifies the Vehicle Identification Number of the vehicle.
16. **Drive** :- Column saving the type of drive of the vehicle.
17. **Size** :- Column specifying the size of the vehicle.
18. **Type**:- Column consisting information about the generic type of vehicle
19. **Paint\_color** :- Column having the color of the vehicle.
20. **Image\_Url** :- URL link to the actual image of the vehicle.
21. **Description** :- Few words describing the condition of the vehicle.
22. **State** :- State in which the vehicle is up for sale.
23. **Lat** :- Latitude of the exact location of vehicle sale.
24. **Long** :- Longitude of the exact location of vehicle sale.
25. **Posting\_date** :- Date when the ad of the vehicle was posted.

**Dataset:**

<https://www.kaggle.com/code/jerrymazeyu/predict-car-price-by-catboost/data>

**RESULT AND ANALYSIS:**

We are aiming to find a model that is more accurate than the others and can better forecast the car pricing value by training and testing multiple models. After all three models have been trained and tested on the dataset, we can observe that XGBoost has the highest accuracy score. It is the most accurate model for estimating the price of a used car.

**CONCLUSION:**

After testing various models, we discovered that the XGBoost is the most accurate model for predicting car prices. We started by cleaning up the data, such as deleting null values and outliers from the dataset. This considerably improves the efficiency of the results.

By making values machine legible and more efficient for training a model, data preparation approaches such as label encoding and normalization assisted to get better results out of the data.

Many data visualization techniques were also investigated in order to learn more about the features and their relationships.

From the above modelling exercise we understand that XGBoost has the highest accuracy level of 91.26%

Hence with such a database where we had to predict the price of used cars provided various predictor attributes, XGBoost delivers the best results.

**REFERENCE:**

1. <https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86>
2. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
3. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>
4. <https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd>
5. <https://www.youtube.com/watch?v=OtD8wVaFm6E>
6. <https://www.youtube.com/watch?v=J4Wdy0Wc_xQ>
7. <https://towardsdatascience.com/what-is-data-cleaning-how-to-process-data-for-analytics-and-machine-learning-modeling-c2afcf4fbf45>
8. https://www.youtube.com/watch?v=-ETQ97mXXF0&t=28115s
9. https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7

**---------------------------------------------END OF DOCUMENT-----------------------------------------**