Analyzing the accelerating sales of used cars in a digitally disrupted market

Mayuravarsha P
Computer Science & Engineering
PES University
Bengaluru, India
mayurvp72@gmail.com

Vedant Mantri
Computer Science & Engineering
PES University
Bangalore, India
vedantmantri3@gmail.com

Pranav Mekal Mahesh
Computer Science & Engineering
PES University
Bangalore, India
pranavmm25@gmail.com

***Abstract*— Used-car selling is being disrupted by the internet revolution, and it's for the better. This new era of digital commerce is more than just about technology; it emphasises the importance of the consumer experience in the used-car purchase process.**

I. Introduction

Online suppliers are beginning to erode traditional used-car dealers' standing and promote growth by empowering digitally savvy customers via three primary capabilities, as indicated by our unique customer research:

- complete end-to-end purchasing capabilities
- extensive vehicle data and photos, along with effective search tools
- unique delivery options

With the development of digital players and the possibility of incumbent-dealer consolidation on the horizon, the developing market will present new dangers and opportunities for firms looking to gain a competitive advantage in an already crowded sector.

Furthermore, while customer purchasing habits are evolving, it is true that the needs of used-car buyers differ significantly from those of new-car buyers. As a result, all used-car merchants must identify their target client categories and quickly design the best ways among a growing number of accessible options in order to provide a uniformly distinctive and distinguishing customer experience.

II. Handling The Dataset

A. Dataset

Our dataset is a used cars dataset chosen from Kaggle. It contains most all relevant information that Craigslist provides on car sales including columns like price, condition, manufacturer, latitude/longitude, and 18 other categories. The link for the same can be found below in references.

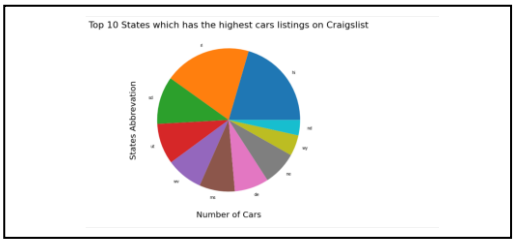
B. Preprocessing or Data Cleaning:

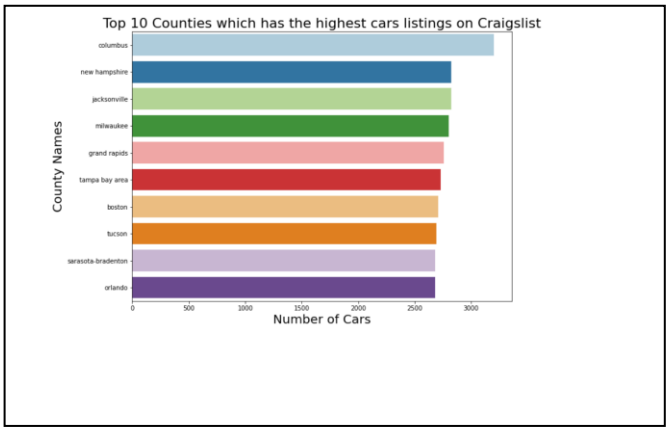
The dataset that we chose was of the size of over 1GB and that means that we have too many redundant columns and data in the dataset and these were removed and dealt with accordingly. The attributes with NULL values we either simply dropped or dealt with using various basic techniques.

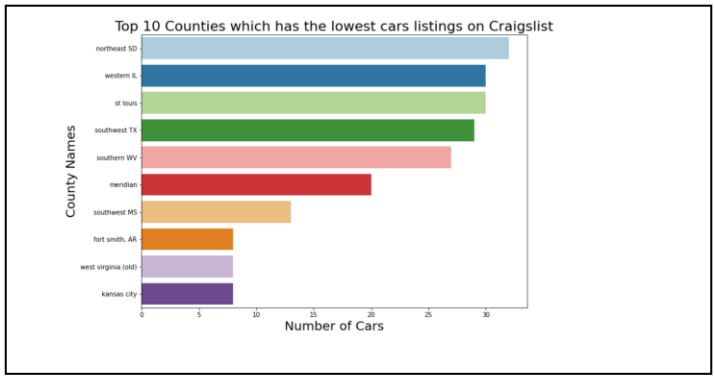
C. Exploratory Data Analysis:

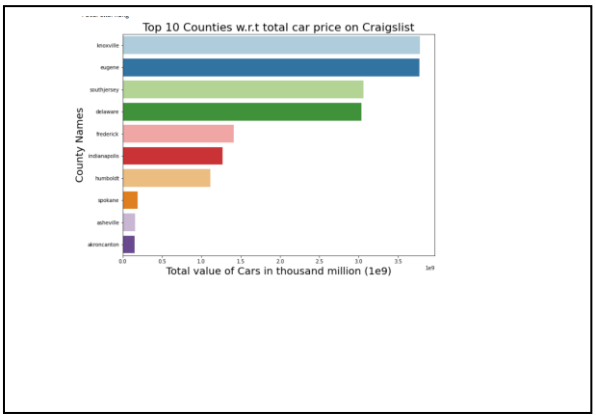
The visual summary of the data makes it easier to identify patterns and trends than looking through thousands of rows and columns on a spreadsheet. Thus, we plotting several data plots and visualization for the better understanding of the data.

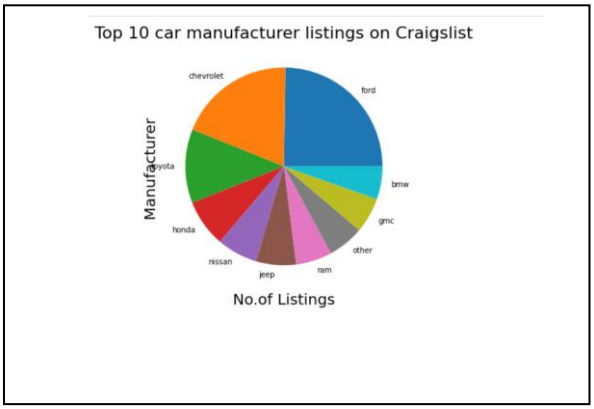
The different plots that we have used are Histograms, Box plots, Count plots and Bar Graphs. Before we plot any graphs, we found that the dataset is huge and might we may face some hitch while doing visualization, so we thought of splitting the dataset and did it randomly.



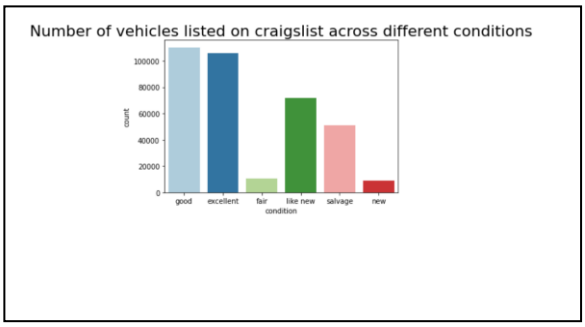








	condition	odometer
0	excellent	109816.415651
1	fair	219937.181593
2	good	89763.987161
3	like new	87025.739294
4	new	36309.288262
5	salvage	244983.404930



III. Existing Appoaches And Their Shortcoming

A. *Linear Regression*

This model does not seem to work well at all as we found results where the RMSE (Root Mean Squared Error) was in the power of 19 and such models are extremely unreliable as they overfit to a huge extent and provide no value in predicting or analyzing. This model assumes that there are only linear relations in the dataset which is a huge gamble and mostly isn't necessarily true and can lead to being extremely costly as mentioned before.

B. *Gradient Boosting*

Does relatively well compared to Linear Regression.

```
linear regression RMSE: 2.2381793934013837e+19
Gradient boosted RMSE: 11488664.771109097
```

IV. Data Modelling

We have built multiple models to predict the likeliness of a used car getting sold based on its various attributes. We followed the standard procedure of cleaning and preprocessing the data before proceeding with model testing. We have used correlation coefficient and root mean square error as the defining basis of which models have performed the best and the worst with the test data. StratifiedKFold from sklearn module has been used to index the training and testing data.

The models used are as follows:

A. *Decision Tree Baseline*

Performs far better than Linear Regression but still doesn't belong anywhere near other machine learning models which have been used by us for better efficiency. The drawbacks of decision trees are pruning and overfitting. There's a high chance of pruning as in most cases we are going to come across outliers and Decision Trees tend to classify each of the data in the training dataset perfectly and hence not only overfitting but also making the length of the tree unnecessarily longer and costlier.

B. *XGBoost Baseline*

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Descent framework. This performs significantly better than Gradient boosting.

C. *XGBoost with Parameters*

The overall parameters have been divided into 3

categories by XGBoost :

General Parameters: Guide the overall functioning

Booster Parameters: Guide the individual booster

(tree/regression) at each step

Learning Task Parameters: Guide the optimization

performed

- This performs significantly better than XGBoost

D. Random forest baseline

Random Forest is an ensemble learning algorithm that constructs many decision trees during the training. It predicts the mode of the classes for classification tasks and mean prediction of trees for regression tasks. It is using random subspace method and bagging during tree construction. It has built-in feature importance. Since it uses ensemble models, it is highly accurate and less error prone compared to previous models.

E. Decision Tree HyperParam Tuning

Hyperparameter tuning is searching the hyperparameter space for a set of values that will optimize your model architecture. Due to this property, it performs way better than a Decision Tree and predicts more accurately.

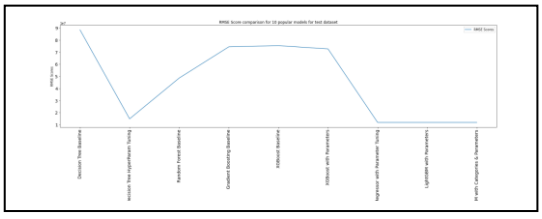
F. Other Efficient Models

- LightGBM with parameters
- MLP Regressor with parameter tuning
- LightGBM with categories & parameters

All of the above-mentioned models are the most efficient and helpful when it comes to analyzing and the proof for the same can be seen below with their r2 and RMSE values.

Comparing all the algorithms:

	r2	rmse
LightGBM with Parameters	-0.00	11767005.41
MLPRegressor with Parameter Tuning	-0.00	11767040.08
LightGBM with Categories & Parameters	-0.00	11767241.05
Decision Tree HyperParam Tuning	-0.57	14728047.11
Random Forest Baseline	-16.00	48518794.41
XGBoost with Parameters	-37.11	72643960.61
Gradient Boosting Baseline	-38.94	74367027.12
XGBoost Baseline	-40.01	75351199.79
Decision Tree Baseline	-55.46	88419103.33



V. Conclusion, Prediction and Analysis

By using an MLP model we saved this model while choosing RELU as the activation function as it performs extremely well in real world examples.

	condition	odometer
0	excellent	109816.415651
1	fair	219937.181593
2	good	89763.987161
3	like new	87025.739294
4	new	36309.288262
5	salvage	244983.404930

The MLP model is saved for future insights on the same

```
# Rerunning MLP Neural Network to save the model
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import GridSearchCV


mlp = MLPRegressor()
param_grid = {
#               'hidden_layer_sizes': [i for i in range(2,20)],
              'activation': ['relu'],
              'solver': ['adam'],
#               'learning_rate': ['constant'],
#               'learning_rate_init': [0.01],
#               'power_t': [0.5],
#               'alpha': [0.0001],
#               'max_iter': [1000],
              'early_stopping': [True],
              'warm_start': [False]
}
model = GridSearchCV(mlp, param_grid=param_grid,
                     cv=10, pre_dispatch='2*n_jobs')

model.fit(X_train, y_train)
```

```
GridSearchCV(cv=10, estimator=MLPRegressor(),
             param_grid={'activation': ['relu'], 'early_stopping': [True],
                         'solver': ['adam'], 'warm_start': [False]})
```

The predictions made for the same can be seen below

```
# Save the neural network model
from joblib import dump, load

filename = 'mlp_neural_network_001.joblib'
with open(filename, 'wb') as file:
    dump(model, file)
# Predict
y_pred = model.predict(X_test)
df1 = pd.DataFrame({"y":y_test,"y_pred":y_pred })
df1.head(50)
```

	y	y_pred
0	15277	38019.127538
1	15108	38399.143017
2	20870	54938.611745
3	3525	35042.930715
4	12654	62525.946993
5	4177	59303.254944
6	4425	32694.897097
7	15897	63461.107457
8	23663	53870.602710
9	30292	61032.725386
10	28877	33493.883099
11	12307	59814.288409
12	99999999	55034.605400
13	5225	48166.994096

ACKNOWLEDGMENT

We would first like to thank our professor Dr. Gowri Srinivasa and her Teaching Assistants for guiding us throughout the course of this project. This work was supported by grants from the Computer Science and Engineering department in PES University, Bengaluru. Everyone from our team have made contributions in all aspects including Data segmentation, Data visualization, and Data modelling analyzing and giving insights and drawing conclusions about the same.

REFERENCES

[1] https://www.mckinsey.com/~/media/McKinsey/Industries/Automotive%20and%20Assembly/Our%20Insights/Used%20cars%20new%20platforms%20Accelerating%20sales%20in%20a%20digitally%20disrupted%20market/Used-cars-new-platforms-Accelerating-sales-in-a-digitally-disrupted-market-vF.pdf

[2] U. Dinesh Kumar - Business Analytics_ The Science of Data-driven Decision Making

[3] Wilkinson, Christopher- Python data science: an ultimate guide for beginners to learn fundamentals of data science using Python

[4] Smith, James, V- Data Analytics: What Every Business Must Know About Big Data And Data Science

[5] https://www.kaggle.com/austinreese/craigslist-carstrucks-data