Car Price Prediction

Submitted by:

Laxmi Narayan

Acknowledgement

I would like to express my gratitude to my primary SME, **Keshav Bansal**, who guided me throughout this project. I would also like to thank my friends and family who supported me and offered deep insight into the study. I wish to acknowledge the help provided by the technical and support staff in the **Data Science of Flib Robo Technologies**. I would also like to show my deep appreciation to my supervisors who helped me finalize my project.

Introduction

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle's price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models across cities in the United States. Our results show that Random Forest model and K-Means clustering with linear regression yield the best results, but are compute heavy. Conventional linear regression also yielded satisfactory results, with the advantage of a significantly lower training time in comparison to the aforementioned methods.

Deciding whether a used car is worth the posted price when you see listings online can be difficult. Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a used car appropriately[2-3]. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.

Machine learning uses data science and makes it feasible to generate such models which are able to make accurate predictions,classisfy things into categories,interpretation of images,etc.This makes the machines and robots intelligent as they can learn on their own and there is no need to worry about their accuracy.In today‟s world,where eveything is getting automated,there surely is

need of the mechanisms which can be eaily trusted for their accuracy.Machine learning along with data science makes this possible.Machine learning is already helping several industries and is well praised for easing the human effort.

Machine learning is a subset of artificial intelligence (AI) wherein algorithms research by way of instance from historical statistics to expect results and uncover patterns which humans cannot spot easily. For instance,ML can screen clients who are probably to churn, possibly fraudulent coverage claims,etc. While ML has been round since the Fifties, latest breakthroughs in lowvalue compute assets like cloud storage, less complicated collection of data, and the proliferation of information science have made it very a good deal “the next huge thing” in commercial enterprise analytics.

Methodology

We utilized several classic and state-of-the-art methods, including ensemble learning techniques, with a 80% - 20% split for the training and test data. To reduce the time required for training, we used 500 thousand examples from our dataset. Linear Regression, Random Forest and Gradient Boost were our baseline methods. For most of the model implementations, the open-source Scikit-Learn package [7] was used.

1. Linear Regression

Linear Regression was chosen as the first model due to its simplicity and comparatively small training time. The features, without any feature mapping,

were used directly as the feature vectors. No regularization was used since the results clearly showed low variance.

2. Random Forest

Random Forest is an ensemble learning based regression model. It uses a model called decision tree, specifically as the name suggests, multiple decision trees to generate the ensemble model which collectively produces a prediction. The benefit of this model is that the trees are produced in parallel and are relatively uncorrelated, thus producing good results as each tree is not prone to individual errors of other trees. This uncorrelated behavior is partly ensured by the use of Bootstrap Aggregation or bagging providing the randomness required to produce robust and uncorrelated trees. This model was hence chosen to account for the large number of features in the dataset and compare a bagging technique with the following gradient boosting methods

SYSTEM DEVELOPMENT

The system requirements for the algorithms to run efficiently and for the implementation of the
whole idea are:

- Windows 10 (64-bit)
- 8 GBRAM
- Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
- ANACONDA
- Python

Performance Analysis

The main reason for this huge market is that when you buy a New Car and sale it just another day without any default on it, the price of car reduces by 30%.

There are also many frauds in the market who not only sale wrong but also they could mislead to wrong price.

So, here I used this following dataset to Predict the price of any used car.

```
#Importing necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import warnings
warnings.filterwarnings('ignore')
```

The next step is to import the training data set in the object named train and the testing data set in the object named test.
This import is done by pandas method of reading .csv files into the system.

```
train_data = pd.read_csv('train-data.csv')
test_data = pd.read_csv('test-data.csv')
```

Next we check the the features present in the loaded data sets.

```
# Looking ate the unique values of Categorical Features
print(train_data['Location'].unique())
print(train_data['Fuel_Type'].unique())
print(train_data['Transmission'].unique())
print(train_data['Owner_Type'].unique())

#Rest Feature are worked for Feature Engineering
```

In Data Analysis We will try to Find out the below stuff

1. Missing Values in the dataset.

2. All the Numerical variables and Distribution of the numerical variables.

3. Categorical Variables

4. Outliers

5. Relationship between an independent and dependent feature(*selling_price*)

Feature Engineering

You can see what the data looks like, but before using it we need to customize it. We will be performing all the below steps in Feature Engineering

- Handling of Missing values

we'll drop the column if it's not serving in our model or we can use **central tendency measures such as mean, median, or mode** of the **numeric feature.**

- **Substitution of Categorical variables**

To be able to use categorical data, we used **pd.get_dummies()** also called **one hot encoding.** This essentially turns every unique value of a variable into its own binary variable.

Model Building

Now,the development of a model for predicting if the user will apply for a loan or not will start.

Dummies will be used for converting categorical variables into numerical variables because sklearn models allows only numerical inputs.

The train data set will be divided into two parts,80% of the data will act as training data and the remaining 20% data will be the validation data.

First we are spliting the data to train and test for the model

```
: from sklearn.model_selection import train_test_split
  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 25)
```

Logistic Regression:

We will first build a Logistic Regression model since logistic regression is used for classification problems.

```
from sklearn.linear_model import LinearRegression
linear_reg = LinearRegression()
linear_reg.fit(X_train, y_train)
y_pred= linear_reg.predict(X_test)
print("Accuracy on Traing set: ",linear_reg.score(X_train,y_train))
print("Accuracy on Testing set: ",linear_reg.score(X_test,y_test))
```

```
Accuracy on Traing set:  0.7083070284244637
Accuracy on Testing set:  0.6991016530826974
```

Random Forest Algorithm:

Now let us calculate the accuracy using another algorithm called Random Forest.

First we have to set up the model for this algorithm.

```
from sklearn.ensemble import RandomForestRegressor
rf_reg = RandomForestRegressor()
rf_reg.fit(X_train, y_train)
y_pred= rf_reg.predict(X_test)
print("Accuracy on Traing set: ",rf_reg.score(X_train,y_train))
print("Accuracy on Testing set: ",rf_reg.score(X_test,y_test))
```

```
Accuracy on Traing set:  0.985360849381709
Accuracy on Testing set:  0.908195577251492
```

Checking accuracy of the model

Evaluating the model accuracy is an essential part of the process of creating machine learning models to describe how well the model is performing in its

predictions. The MSE, MAE, and RMSE metrics are mainly used to evaluate the prediction error rates and model performance in regression analysis.

- **MAE** (Mean absolute error) represents the difference between the original and predicted values extracted by averaged the absolute difference over the data set.
- **MSE** (Mean Squared Error) represents the difference between the original and predicted values extracted by squared the average difference over the data set.
- **RMSE** (Root Mean Squared Error) is the error rate by the square root of MSE.

```
from sklearn import metrics
from sklearn.metrics import mean_squared_error, mean_absolute_error

print("\t\tError Table")
print('Mean Absolute Error      : ', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared  Error      : ', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared  Error : ', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('R Squared Error          : ', metrics.r2_score(y_test, y_pred))
```

```
               Error Table
Mean Absolute Error      :  1.519392098277609
Mean Squared  Error      :  10.758598828030518
Root Mean Squared  Error :  3.2800303090109577
R Squared Error          :  0.908195577251492
```

Future Work

For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset. To correct for overfitting in Random Forest, different selections of features and number of trees will be tested to check for change in performance.

References

1. https://www.kaggle.com/jpayne/852k-used-car-listings

2. N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya and P. Boonpou, "Prediction of prices for used car by using regression models," 2018 5th International Conference on Business and Industrial Research (ICBIR), Bangkok, 2018, pp. 115-119.

3. Listiani M. 2009. Support Vector Regression Analysis for Price Prediction in a Car Leasing Application. Master Thesis. Hamburg University of Technology

4. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.

5. Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in Neural Information Processing Systems. 2017.

6. Fisher, Walter D. "On grouping for maximum homogeneity." Journal of the American statistical Association 53.284 (1958): 789-798.

7. https://scikit-learn.org/stable/modules/classes.html: Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011