Car Price Predection Model using Python

ENROLMENT - 20103282, 20103296, 20103297

NAME OF STUDENTS – ARYAN DHAOR, YASH KAPOOR, KUSH KAPOOR

NAME OF THE SUPERVISOR – MR. JANARDAN VERMA

OSS PROJECT 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY
NOIDA, UTTAR PRADESH

CONTENT OF SYNOPSIS

PROBLEM STATEMENT

The research objective of this study is to predict used cars prices using Machine Learning concepts, by scraping data from websites that sell used cars, and analysing the different aspects and factors that lead to the actual used car price valuation. To enable consumers to know the actual worth of their car or desired car, by simply providing the program with a set of attributes from the desired car to predict the car price. The purpose of this study is to understand and evaluate used car prices, and to develop a strategy that utilizes Machine Learning concepts to predict used car prices.

NEED

This project aims to deliver price prediction models to the public, to help guide the individuals looking to buy or sell cars and to give them a better insight into the automotive sector. Buying a used car from a dealer can be a frustrating and an unsatisfying experience as some dealers are known to deploy deceitful sale tactics to close a deal. Therefore, to help consumers avoid falling victims to such tactics, this study hopes to equip consumers with right tools to guide them in their shopping experience.

OBJECTIVE

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.

METHODOLOGY

We utilized several classic and state-of-the-art methods, including ensemble learning techniques, with a 90% - 10% 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 [1] 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.

3. Gradient Boost

Gradient Boosting is another decision tree based method that is generally described as “a method of transforming weak learners into strong learners”. This means that like a typical boosting method, observations are assigned different weights and based on certain metrics, the weights of difficult to predict observations are increased and then fed into another tree to be trained. This model was chosen to account for non-linear relationships between the features and predicted price, by splitting the data into 100 regions.

4. XGBoost

Extreme Gradient Boosting or XGBoost [4] is one of the most popular machine learning models in current times. XGBoost is quite similar at the core to the original gradient boosting algorithm but features many additive features that significantly improve its performance such as built in support for regularization, parallel processing as well as giving additional hyperparameters to tune such as tree pruning, sub sampling and number of decision trees. A maximum depth of 16 was used and the algorithm was run

on all cores in parallel.

TECHNOLOGIES USED

Languages : Python

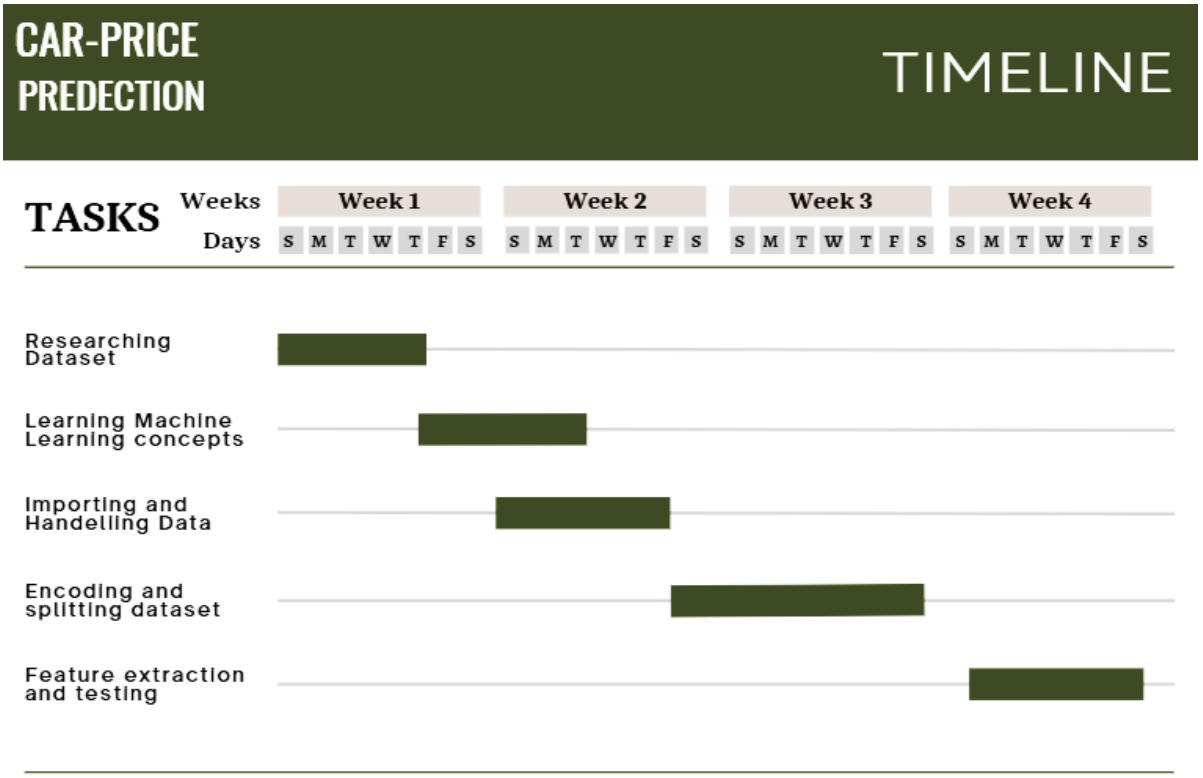
Operating System : Windows

Softwares : VS Code, Anaconda Navigator , Jupyter, SPyder

Python Libraries : Numpy, Pandas, MatPlotlib, Scikit, Requests,

Internet resources for Dataset

TIMELINE



REFERENCES

- [1] https://www.javatpoint.com/data-preprocessing-machine-learning
- [2] https://www.kaggle.com/datasets/hellbuoy/car-price-prediction
- [3] https://www.kaggle.com/datasets/jpayne/852k-used-car-listings
- [4] https://www.analyticsvidhya.com/blog/2021/07/car-price-prediction-machine-learning-vs-deep-learning/