MINOR PROJECT 1

REPORT ON

MACHINE LEARNING PIPELINE ON CLOUD

SUBMITTED BY:

Aditya Kumar	Anuj Verma	Dhananjai Kalra	Mayank Joshi
500066319	500069910	500070904	500070105

UNDER THE GUIDANCE OF

GAGAN DEEP SINGH
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE

DEPARTMENT OF COMPUTER APPLICATION
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN-248007
2020

DEPARTMENT OF COMPUTER APPLICATION
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES, DEHRADUN

MINOR PROJECT 1

Project Title: Predictive Analysis using Machine Learning

Abstract:

The need for new technology such as Machine Learning and Data Science is on a rise today. Whilst technology was just starting to grow a decade or 2 earlier, the methods, tools and algorithms developers created then are still being used today. And although these techniques were effective then, today it is a completely different scenario. Machines of this era have reached their limited potential. They can only get involved when they start to think for themselves. With this project we are embarking on this colossal feat of helping machines learn.

Our project is about developing a reselling platform for cars and hosting it on the cloud. This platform will be developed using python and will use machine learning algorithms to suggest best recommendations to the user. The project also consists of the elements of website development which will provide a user interface to our web app. It will ask for a significant amount of data regarding the selling points and price range of a vehicle. Based on this data and certain results through statistical and graphical analysis, the machine learning algorithm will predict the required output.

Keywords: Machine Learning, Regression, Python Programming, Feature Importance, Feature Engineering, Visualization, Flask Framework, Cloud Platform, Web App, Version Control System

Introduction:

The prospect of technology, society and change has been subjected to many contradictions over the years. Needless to say, over the past, these orthodox differences have been replaced with acceptance and harmony. Finally, with that outreach, we realize that the social climate of dependence over the machines today is not just a result of the changing belief but also a result of opportunities that they can bring to the table.

Now, arguing over their social effects is something left for the philosophers to discuss, we as the practitioners of the field thrive to bring new and more advanced contraptions to achieve something greater. And one such vibrant call is the pursuit of Machine Learning.

Machine Learning or ML, as it is known commonly, is the future. It is exactly as it sounds, giving machines the ability to learn, adapt and change accordingly. It might sound easy, but machine learning is quite a feat to accomplish. For years now developers and researchers have been busting their heads on this topic. And although we have found some success, we still have a long way to go.

One of the applications of machine learning is predictive analysis. Certain modules and algorithms implemented through Python Programming make this task comprehensible. The machine can learn the users' previous choices and tends to make recommendations based upon what is learned. It's a simple mechanism on the outside, but when one looks at the big picture it shows us that you are granting the ability to think and act to an entity that has been constructed out of screws, wires and chips. It's almost magical and yet so attainable. Machine learning is the future of science, the future of computers, and the future of our modern vibrant society.

Problem Statement:

People require vehicles and they want them cheap and in the best condition possible. Simply because of that need, they go to dealers and websites trying to purchase second-hand vehicles for cheaper prices. Where the dealers will make a fool out of these customers on the face, websites do it by running a restless algorithm which instead of best prices and new products shows the same results again and again to different users. Now that is a problem because who doesn't like variety? Who doesn't want excessive choices leading to descriptive purchase? With these questions in mind, the need for a new and efficient practice based on machine learning seems of importance.

Literature Review:

The goal of this project is to develop a platform over cloud for predicting prices of second-hand vehicles. As such, the platform will use a machine learning algorithm to perform predictive analysis. By applying ML for predicting prices of used vehicles we are speeding the process of buying cars directly from sellers and reducing the middle broker involvement.

The ML Pipeline will be created using python and the following libraries/modules:

1. pandas
2. numpy
3. seaborn
4. matplotlib
5. sklearn
6. pickle

All these modules together offer the quality of machine learning to our algorithm. Using statistics, analytics and visualizations, we will create a well-defined and descriptive view of the data over a graph which makes it easier for users to make sense out of mere numbers.

Our algorithm, in the end, would suggest the best choices based upon these plots and on users' needs demands.

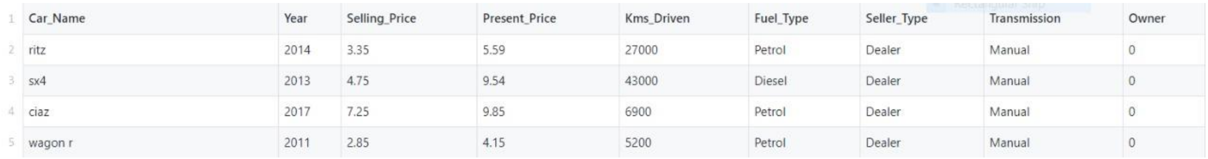
Not only that but everything from selection to searching to suggestion all will be done over cloud. This means that users won’t be bothered to keep any unnecessary data with them on their system. Also, using cloud the working of this ML algorithm will be much faster, cheaper and efficient.

This machine learning model will generate a binary file and will then be used as a backend service for our web application.

Methodology:

Dataset:

The dataset used is a sample of several vehicles from differing brands contrasting in different qualities. A sample of the dataset is provided below



Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0

The dataset used in this project is flexible and can be replaced with a much bigger dataset of the same sort. It has the following features:

a. Car_name: Name and branding of the car

b. Year: Year of the first purchase

c. Selling_price: price at which sold

d. Present_price: price when purchased

e. Kms_driven: How much distance the car has travelled

f. Seller_type: who is selling

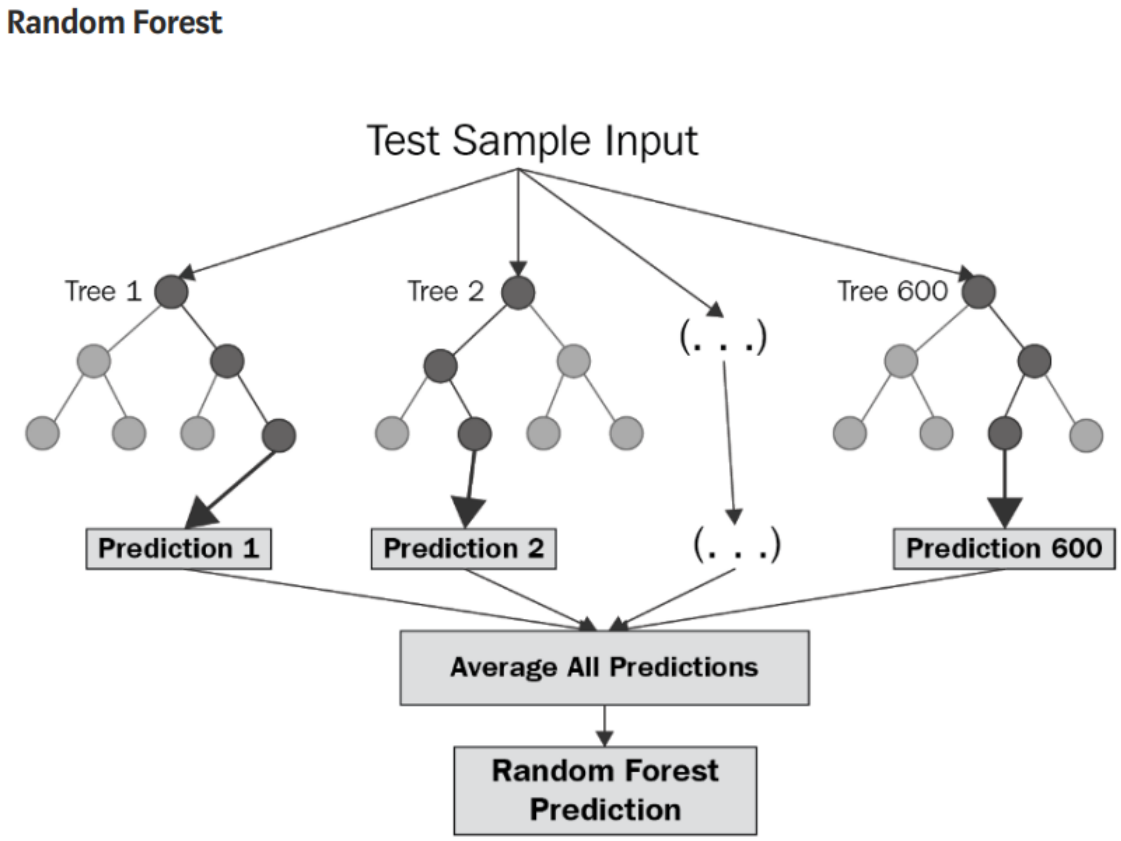
g. Transmission: Manual/Auto

h. Owner: number of previous owners car had(0 qualifies for second-hand)

In the dataset there are 302 example vectors which were all taken from Kaggle. Kaggle is a machine learning and data science community which provides free datasets. As mentioned before, the dataset used here is flexible and susceptible to change. That is, if a larger dataset has to be used, the ML algorithm will take time to learn the new entries but will also work for the same as long as the variables are in order.

ML Algorithm:

We have used the Random Forest Regressor model for Machine Learning. It is an Ensemble technique which is capable of performing both regression and classification tasks by combining multiple Decision Trees together by a technique called Bagging. Bagging, in Random Forest method, involves training each Decision Tree on a different data sample where sampling is done with replacement. Ensemble technique is used for combining multiple decision trees rather than relying on multiple individual decision trees. Bagging method is used for reducing variance for algorithms with high variance, like Decision Trees. Bagging makes each model run independently and then aggregates the final output at the end.



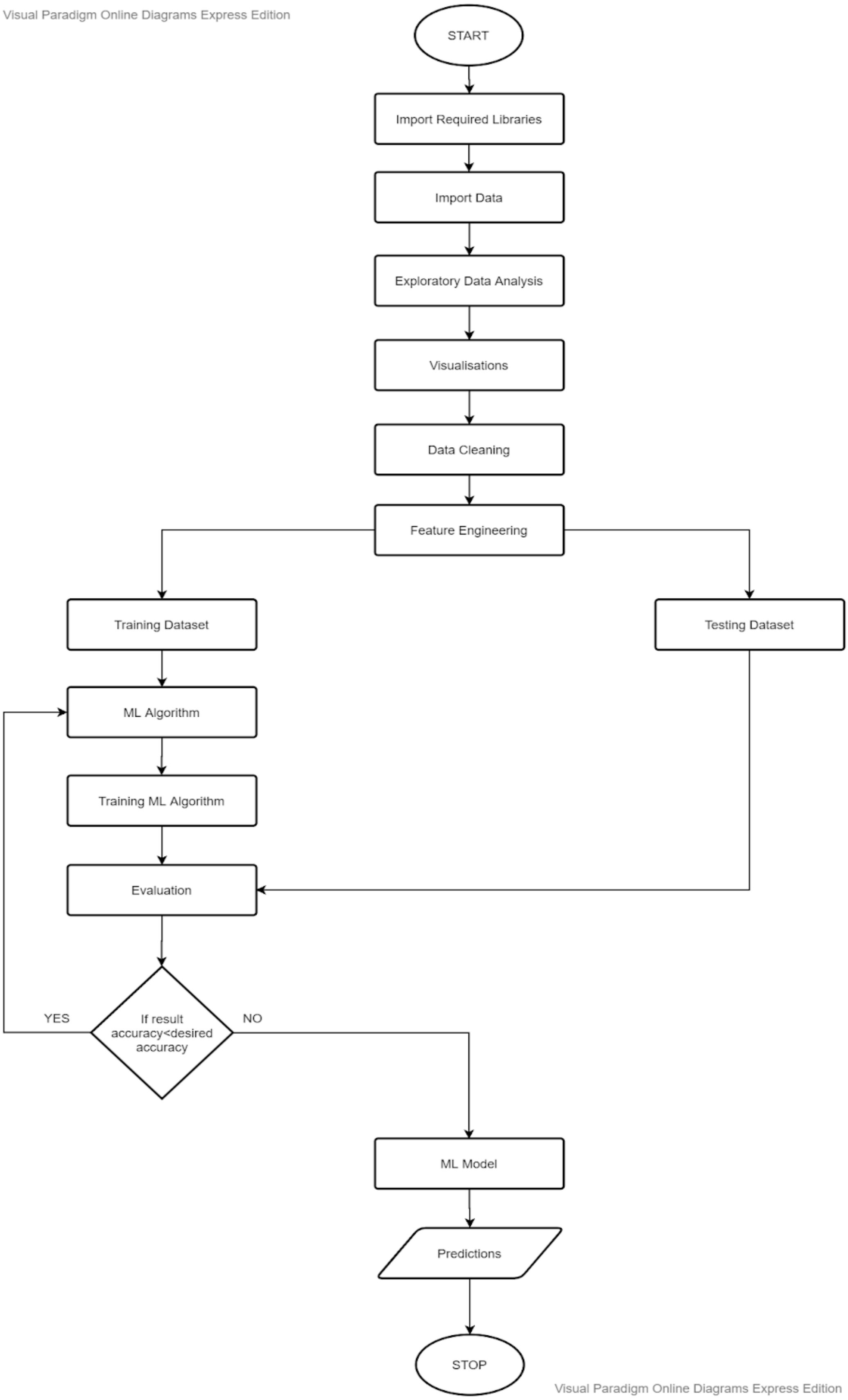
Random Forest Structure

In a Random Forest Regressor, samples are taken repeatedly from the training data so that each data point has an equal opportunity of getting selected.

Parameters of Random Forest Regressor:

1. n_estimators: This denotes the number of trees in the random forest.
2. max_features: The number of features to look for best split
3. max_depth: The maximum depth of the tree
4. min_samples_split: The minimum number of samples required to split an internal node
5. min_samples_leaf: The minimum number of samples required to be at a leaf node

Flow Chart: (Machine Learning Model)



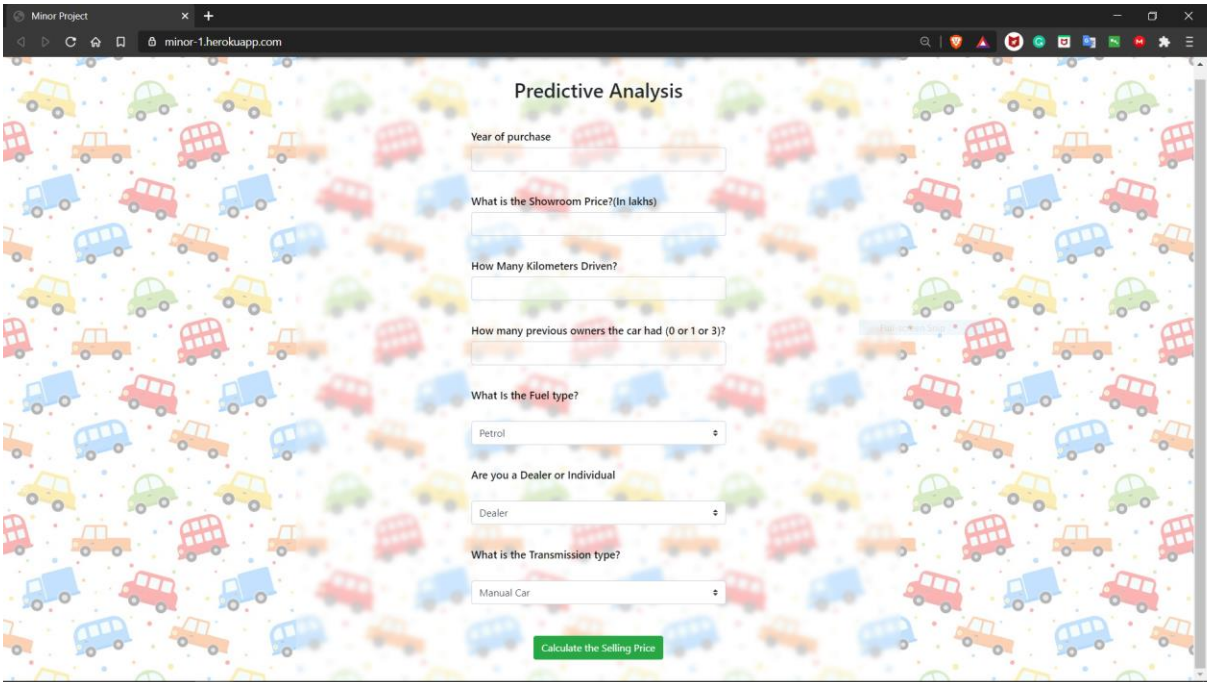
Visual Paradigm Online Diagrams Express Edition
START
Import Required Libraries
Import Data
Exploratory Data Analysis
Visualisations
Data Cleaning
Feature Engineering
Training Dataset
Testing Dataset
ML Algorithm
Training ML Algorithm
Evaluation
YES
If result accuracy<desired accuracy
NO
ML Model
Predictions
STOP
Visual Paradigm Online Diagrams Express Edition

Web Application:

According to a study conducted by ML.India (Machine Learning India),

"90% of ML models cooked up by scientists, students and other practitioners never actually make it to production"

Although this might be true for most of the projects in the field, our approach to this problem of deployment was quite optimistic. We created a very simple web app which was both user friendly and presentable. The Web App was designed using Python, HTML, and CSS and it is deployed on Heroku Cloud. Python's Flask Framework was used to develop the backend for Web Application.



Predictive Analysis

Year of purchase

What is the Showroom Price?(In lakhs)

How Many Kilometers Driven?

How many previous owners the car had (0 or 1 or 3)?

What Is the Fuel type?

Petrol

Are you a Dealer or Individual

Dealer

What is the Transmission type?

Manual Car

Web Application UI

Features and advantages of the web app are listed below-:

1. It is very easy to navigate with an intricate user friendly design scheme.

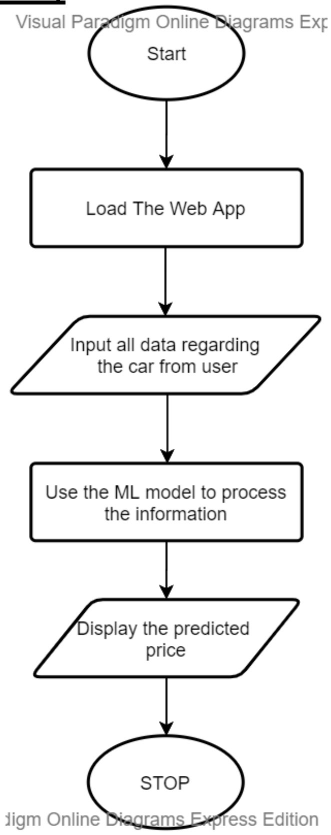
2. As seen in the image, it features several input fields asking necessary details on which the predictive analysis will be conducted.

3. Hosting on Heroku Cloud has ensured enhanced performance through rich application monitoring.

4. Furthermore, the Web App just like our data set is scalable. Flask gives our Web App the required scalability, flexibility and modularity which it might require in the near future. The design can be tweaked without hindering the performance.

5. As it is all developed in Python, the developers have a much easier time in maintaining the code and adding new features.

Flow Chart: (Web Application)


Visual Paradigm Online Diagrams Exp
Start
Load The Web App
Input all data regarding the car from user
Use the ML model to process the information
Display the predicted price
STOP
digm Online Diagrams Express Edition

Objectives:

1. Predicting prices of used vehicles for resale using predictive analysis and machine learning.

2. Developing and deploying a web-app on the cloud to make it accessible to the users and help them predict prices based on their required features.

3. Python for Developing ML Model: ML algorithm and the development of the whole platform will be done using python. Python and its different modules have been considered very effective for practising machine learning. This makes python a suitable development tool for this platform. All the analytics, statistics and visualizations is done using python programming.

4. Flask for Developing Web Application: The Flask Framework will be used for developing the web application for our project which then will be deployed on a cloud service.

5. Cloud Services for Code Hosting and Web App Deployment: The use of the cloud is one of the key components of this project. It is fast, it is cheap, it is reliable and the best of it all is that it is the most secure connection between the service provider and the consumer. Cloud will loosen the burden on the user’s end and at the same time grant developers or the institution at hand safe encapsulation of data.

Advantages:

This project is a collaboration of several different elements of machine learning and web development. Now, here in this section we will throw some light on the advantages of this collaborative effort.

1. Ml algorithms are susceptible to overfitting. This means, good performance on the training data and poor performance on the new data. Using random forest regression we are averaging several trees, which significantly reduce the chances of overfitting.

2. On prediction, it becomes very easy to measure the relative importance of each feature. Feature importance matrix is a powerful library of Sk-Learn which features a nice way of getting insights on data.

3. RFS has very few statistical assumptions. It does not assume that data is normally distributed, linear, or any specified interactions.

4. It’s default hyper parameters often produce a good prediction result, therefore, making it easier to use. Also, it has an in-built validation mechanism named Out-of-bag (OOB) score.

5. For the most part the ML algorithm is the most important aspect of this project. But the use of flask framework and Heroku cloud for web development and deployment of the web app cannot be neglected. Using these web development tools brought modularity and scalability to our project and made deployment a much simpler task.

Conclusion:

We conclude that for this project of machine learning for predictive analysis, the random forest regression model that we chose has proven to be the right way to go. We had a dataset containing details of several different vehicles and we were attempting to use this information to predict the best reselling value of these vehicles. The goal was to create an ML algorithm which will predict the reselling amount to 99% accuracy. In turn, we were able to create an ML algorithm which is accurate and flexible to its core. This project brandishes features like scalability and modularity. Not only machine learning but web development was also an integral part of this project. In future, if a larger dataset arrives or further studies on this field are conducted, then this project can be used as a reference tool for such endeavours.

References:

1. **Dataset:** https://www.kaggle.com/adityakumaar/vehicle-price-prediction?select=vehicle_dataset.csv
2. **Sklearn random forest regressor**: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
3. **Seaborn library**: https://seaborn.pydata.org/
4. **Matplotlib library**: https://matplotlib.org/
5. **Work flow of ML project**: https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94