

DP Experience Attribution Analysis

Project report submitted in partial fulfilment of the requirement for
the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Rajat (181429)

Under the supervision of

Dr. Amol Vasudeva

To



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology Waknaghat, Solan-
173234, Himachal Pradesh**

Candidate's Declaration

I hereby declare that the work presented in this report entitled “**DP experience attribution analysis**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from February 2022 to May 2022 under the supervision of **Dr. Amol Vasudeva** Assistant Professor (SG) Department of Computer Science And Engineering. The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Rajat, 181429

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr Amol Vasudeva
Assistant Professor (SG)
Department of Computer Science And Engineering.

Acknowledgement

This article is to acknowledge all the people without whom this project would not have been completed. Majorly, I would like to thank our supervisor Dr. Amol Vasudeva who gave his immense support, dedicated his time towards it and made us understand how to make this project. Without his guidance and learning, this work would not have been completed.

The preparation of this project was an immense learning experience and we inculcated many personal qualities during this process like responsibility, punctuality, confidence and others.

We consider, project as a bridge between practical and theoretical learning and with this thinking we worked on the project and made it successful due to timely support and efforts of all who helped me.

Once again, we would like to thank our classmates and our families for their encouragement and help in designing and making our project creative, we are obliged to all of these. Only because of them we are able to create our project and make it a good and enjoyable experience.

Table of Contents

Candidate's Declaration.....	i
Acknowledgement	ii
Table of Contents	iii
List of Abbreviations	v
List of Figures	vi
List of Tables	vii
Abstract.....	viii
1 Introduction.....	1
1.1 Problem Statement	2
1.2 Objectives	3
1.3 Methodology.....	3
1.4 Organization.....	14
2 LITERATURE SURVEY	15
2.1 Survey.....	15
2.2 Existing Applications	15
2.2.1 XGboost	16
2.3 Proposed Application	17
2.4 Feasibility study.....	17
2.4.1 Economic feasibility.....	17
2.4.2 Probability feasibility:-	18
2.4.3 Technical feasibility	18
3 SYSTEM DEVELOPMENT.....	18
3.1 Tools & Technologies Used	19
3.1.1 Python.....	19
3.1.2 Html.....	19
3.1.3 CSS.....	20
3.1.4 STREAMLIT	20
3.1.5 Scikit-Learn	21
3.1.6 Language Used in This Project.....	22
3.2 Requirements for Hardware and Software.....	22
3.3 Functional-Requirements.....	23
3.4 Non-Functional Requirements	24
3.5 Deployment of the model	24
3.6 User Interface Functionalities Requirements	26

3.7	Flow Chart.....	26
3.8	Use Case Diagram.....	27
3.9	Machine Learning.....	28
4	PERFORMANCE ANALYSIS	30
4.1	DS USED IN THE PROJECT.....	30
4.2	DATA SET FEATURES	30
4.2.1	Types of Data Set	30
4.2.2	Number of Attributes, fields, and the data set description.....	31
4.3	Performance evaluation	31
4.4	Various Results and Output At Different Stages and Website UI:	35
5	Conclusion.....	41
5.1	Conclusion	41
5.2	Future Scope.....	42
6	REFERENCES	43
7	APPENDICES	44

List of Abbreviations

DP	Delivery Partner
DPX	Delivery partner experience
DB	Data Base
CSV	Comma Separated Values
EDA	Exploratory Data Analysis
E-commerce	Electronic Commerce
ARIMA	AutoRegressive Integrated Moving Average
Xgboost	Extreme gradient boosting

List of Figures

<i>Figure 1 amazon sales over year</i>	2
<i>Figure 2 Encoders</i>	7
<i>Figure 3 vectorization</i>	7
<i>Figure 4 Inserting</i>	8
Figure 5 Overall score plot.....	12
Figure 6 overall score plot 1	12
Figure 7 retail bank rfm	16
Figure 8 Xgboost	16
<i>Figure 9 DS MODEL</i>	17
Figure 10 streamlit	25
Figure 11 Deployment	25
Figure 12 sharing of app	26
Figure 13 flow chart	27
Figure 14 UC diagram	28
Figure 15 regression output.....	31
Figure 16 classification output	32
Figure 17 AUC	32
Figure 18 Dashboard	36
Figure 19 user interface.....	38
Figure 7.1 Code Snippets	44

List of Tables

Table 3.1 Languages	22
Table 3.2 Libraries	22
Table 3.3 NFR	24

Abstract

We know that we spend a lot of time and effort launching different pilots and programs which may or may not succeed. We have limited scientific mechanisms to tell which area we should focus on and what is important for improving the driver experience.

The main aim of this project is to improve the driver experience. As company continues to become more driver-centric, there is an increasing need for identifying the overall driver experience by better analysing multiple levers around delivery partners for proactive resolution.

Once we know all the key levers, we can have a scientific method to tell why x (attribute) is more important than y (attribute). We can easily tell on changing x how our overall driver experience score impacted. One of the key gaps cited by product teams has been the lack of a single-source platform to analyze the overall delivery partner experience.

We aim is to build a robust scientific framework to identify levers impacting driver experience. So, we can easily approach the driver experience in a more structured, targeted, and scientific manner.

This document details our learnings and proposed next steps.

CHAPTER NO. 1

1 Introduction

So, with increased e-commerce, we need better tools to analyze and track the performance of different kinds of products and a variety of things that are being made handy by our tool. We can track customer experience in a fast manner to make important decisions as soon as possible to survive in the competition of the world.

Improve the DP experience through multiple programs launches and grievance addressal mechanisms. The purpose of this project is to provide a science driven model which helps us identify what lever impact DrEX.

We identified 4 sub-problems to solve this problem:

- [1] Quantifying DP Experience as the dependent variable
- [2] identifying levers and generating hypotheses around their impact on DrEX
- [3] Performing an attribution analysis to check for the impact of these levers on DP experience
- [4] Interpret the results and identify the top n levers which impact the DP experience.

For identifying the levers, firstly we need to generate an overall DP experience score. Then, to analyze whether a particular attribute is impactful in enhancing their experience or not, we have to generate multiple hypotheses. After that, to determine which attribute has the greatest impact on our decision to convert, we perform the attribution analysis and take our desired next steps.

By identifying the top levers, we can determine the rate of investment (ROI) that if we improve lever A by X%, by how much will DrEX improve? is more DP inactive? What will be the overall impact on NPS, CSAT, and CES? Through this we can invest our time and budget on more targeted programs.

Pre analysis done short overview

In order to understand driver's experience, we created dashboards that tells the [1] NPS (Net Promoter Score) [2] CSAT (DP Overall Satisfaction Score) [3] CES (DP Effort Score)

The Net Promoter Score is an index ranging from -100 to 100 that measures the willingness of DP. It is used as a proxy for gauging the DP loyalty to the brand.

CES (Customer Effort score) is used to measure how much effort a DP has to exert to get an issue resolved and how easy was for a DP to interact with the app.

CSAT (Customer Satisfaction), is a key performance indicator used to track how satisfied DPs are with services.

Through this analysis, we measured the overall DP loyalty score or Net Promoter Score (NPS). We calculated the Customer effort score for it. We calculate the CSAT score and we have achieved a CSAT score of some satisfactory. We calculated the DP experience score on the provider id level using different scientific methods.

Worldwide Amazon Retail Ecommerce Sales, 2017-2021
billions and % change

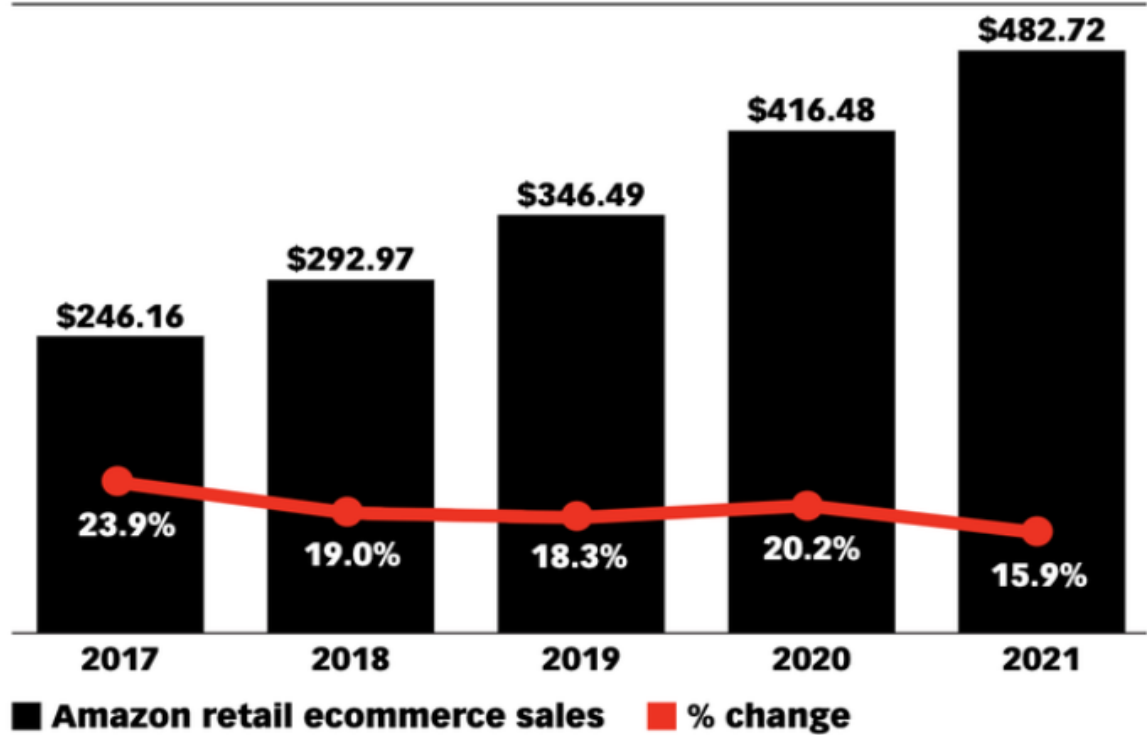


Figure 1 amazon sales over year

1.1 Problem Statement

Improving driver experience is a difficult task and we have multiple programs launches and grievance addressal mechanisms. With the pool of DP experience data, we have we want to identify the top n levers which has the most impact of DP experience.

This is an open-ended problem statement which requires the following exercises:

- [1] Quantifying DP experience as dependent variable.
- [2] Generating multiple hypothesis around levers that impact DP experience.
- [3] Performing an attribution analysis to check for impact of these levers on DP experience.
- [4] Interpret the results and identify the top N levers which impacts DP experience.



1.2 Objectives

The purpose behind DP experience attribution **analysis tool is to predict the experience of a delivery partner** and predicting on large datasets to have an efficient solution.

We can easily tell the probability score of a good, neutral or bad experiences of the driver by using XG boost algorithms.

It is a full stack deployed project in which we have to give inputs based on some parameters which helps us to predict the output.

Main objectives are:

- >This Web app helps user to predict the delivery partner experience. 
- >It turns your excel sheet into a full fledge label generated score of delivery partner.
- >Analysing millions of rows become easy. 
- >This Web app allows you to do Exploratory data analysis.

1.3 Methodology

The major work of a project is to mainly

- [1] Quantifying the DP experience as Dependent variables
- [2] Identify the top n levers which impact the DP experience
- [3] Do pre-analysis and run an attribution model.

At the end of this project, we were able to run our xgboost classification model and able to find out the accuracy of the model we ran. We can do correlation and regression analysis.

[1] For quantifying the DP experience we went through the DP journey and also spent some time with DPs to find out their pain points and used this information to identify the independent variables. By doing a lot of research and by team collaboration, we found all

the touchpoints of DP through which DPs express their experience [1] A source [2] B source [3] C source [4] D source [5] E source [6] F source. We were not able to include all these levers because of limitation in data availability and total data volume.

We took only three sources to quantify the DP experience: -

[1] A source - In A source, questions have been asked for each DP have to answer them, for which we get their response score between (1 to 5) where 1 is marked as least satisfied and 5 is marked as most satisfied.

[2] B source- Different surveys will be conducted by the team for getting B sources feedback from the drivers.

[3] C source- It include drivers who are not vocal about their experiences through these surveys, A source.

How we processed A source data.

For processing, we selected particular questions from each category.

After selecting all the questions, we wrote SQL queries on the and got results, who did respond to a given question. DPs who did not answer some particular questions had missing values associated with them. After getting results in the cloud desktop, we wrote python scripts for data cleaning through outlier removal. We found that there are more than 40% missing values present in the data for each question, so for filling those missing values we find two ways [1] KNN imputer (which took around 1 hour) [2] Missing at random iterative imputer.

We rejected Missing at random imputer because in this approach we fill the missing value by the mean of the columns and then predict missing values by using other columns.

So, we applied KNN imputers for A source which fills the missing values of response score based on other DPs with similar behaviours.

Then we took the average of all the response scores and were successfully able to get the overall A source score for each DP between the range 1-5.

How we processed B source.

Similarly, for processing the other sources we wrote SQL queries for fetching the sentiments given by the drivers. For gauge the sentiments, we researched on Natural Language Processing models and settle with the BERT pre-trained neural network model for generating sentiments in the end. We used the BERT model because that it has the highest training accuracy of 99.7% on training datasets and 93.8% accuracy of validation datasets.

We wrote python scripts for generating the overall sentiment using the BERT pre-trained neural network model and were successfully able to generate a score between 1 and 5. This whole process takes 4.5 hours to generate sentiment for DP's. We used the vaex python library which converts our dataset in HDF5 format (hierarchal data file format 5/ small chunks of metadata) for generating sentiment score and in order to further reduce the time

taken by BERT we used pyspark. The time taken to generate all sentiments was reduced from 4.5 hours to less than 1 hour.

How we processed the final analysis.

Now, we have an overall A source and B source score. We also find the dp which presents in both the sources. But we can't lose data for the remaining DPs because the feedback given by the DPs shows the actual experience of the DPs. So, we building on the hypothesis that a DP with good experience will have higher engagement. It was connected to their experience. We leveraged engagement scores which is built on engagement metrics as an input.

We leveraged the engagement score to fill in for the non-vocal DPs (i.e., the DPs who are neither in A source nor in the B source). Now to generate the final experience score for each DPs we leveraged the sources using the class weights method i.e., the class which is in the minority gets the higher weight and the class that is in the majority get the lower weight. Then we checked the normal distribution curve on the final scores we generated and find right skewness in the data. To check the skewness, we have made Q-Q plots and explored transformation techniques like logarithmic transformation, exponential transformation, and box-cox transformation. Finally, we identified box-cox transformation as final as it I the best fit This is important so that there will be no biases in the data.

So, we are successfully able to quantify the DP experience score for each DPs.

After getting all successful results we researched a lot and found 16 other independent variables which is controllable or non-controllable and build a whole dataset by writing 16 different sql queries. We use pycaret library to quickly ramp up on the different models for doing analysis purpose in a quick manner.

We found in our analysis that the regression models do not fit the data correctly and we got a goodness of it less than 10 percent so we divide the dp experience score in a 3 buckets and turns our regression model into classification model so that we can achieve higher accuracy. We compare more than 15 models and done our analysis on xg boost model with around 65 percent of accuracy. We found the variable importance to tell which variable is more important to predict the driver experience in more accurately manner.

More about BERT

Steps involved in Bert Neural Network: -

The first step in processing text is to cut it into pieces, called tokens. There are many variations of how to do it, but BERT uses word piece tokenization. This means that tokens correspond roughly to words and punctuation although a word can also be split

into several tokens if it contains a common prefix or suffix. Words can even be spelled out if they have never been seen before.

The second step is to associate each token with an embedding, which is nothing more than a vector of real numbers. Again, there are many ways to create embedding vectors. Fortunately, already trained embedding is often provided by research groups, and we can just use an existing dictionary to convert the word piece tokens into embedding vectors.

The embedding of tokens into vectors is an achievement in itself: The values inside an embedding carry information about the meaning of the token, but they are also arranged in such a way that one can perform mathematical operations on them, which correspond to semantic changes like changing the gender of a noun, or the tense of a verb or even the homeland of a city. However, embedding is associated with tokens by a straight dictionary lookup, which means that the same token always gets the same embedding, regardless of its context.

The third step is where the attention mechanism comes in, and specifically for BERT the scaled dot-product self-attention. Attention transforms the default embeddings by analysing the whole sequence of tokens so the values are more representative of the token they represent in the context of the sentence.

How BERT take care of pre-processing?

[1] stemming or lemmatization: Bert uses BPE (Byte- Pair Encoding to shrink its vocab size), so words like run and running will ultimately be decoded to run + ##ing. So, it's better not to convert *running* into *a run* because, in some NLP problems, you need that information.

[2] De-Capitalization - Bert provides two models (lowercase and uppercase). One converts your sentence into lowercase, and others will not change related to the capitalization of your sentence. So you don't have to do any changes here just select the model for your use case.

[3] Removing high-frequency words - Bert uses the Transformer model, which works on the attention principle. So, when you finetune it on any problem, it will look only at those words which will impact the output and not on words that are common in all data.

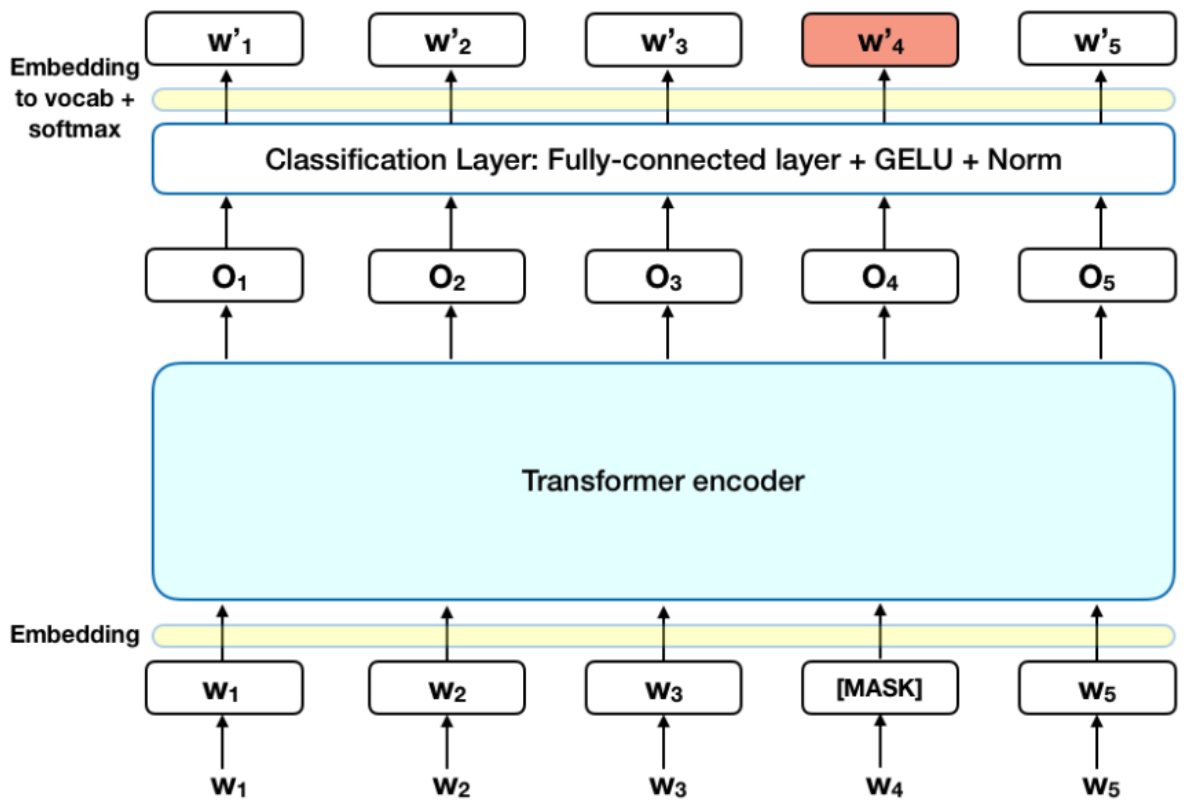


Figure 2 Encoders

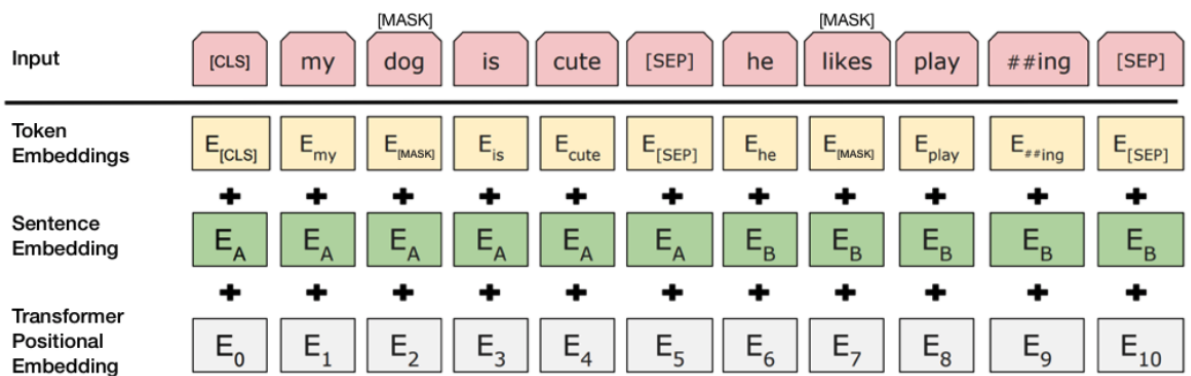


Figure 3 vectorization

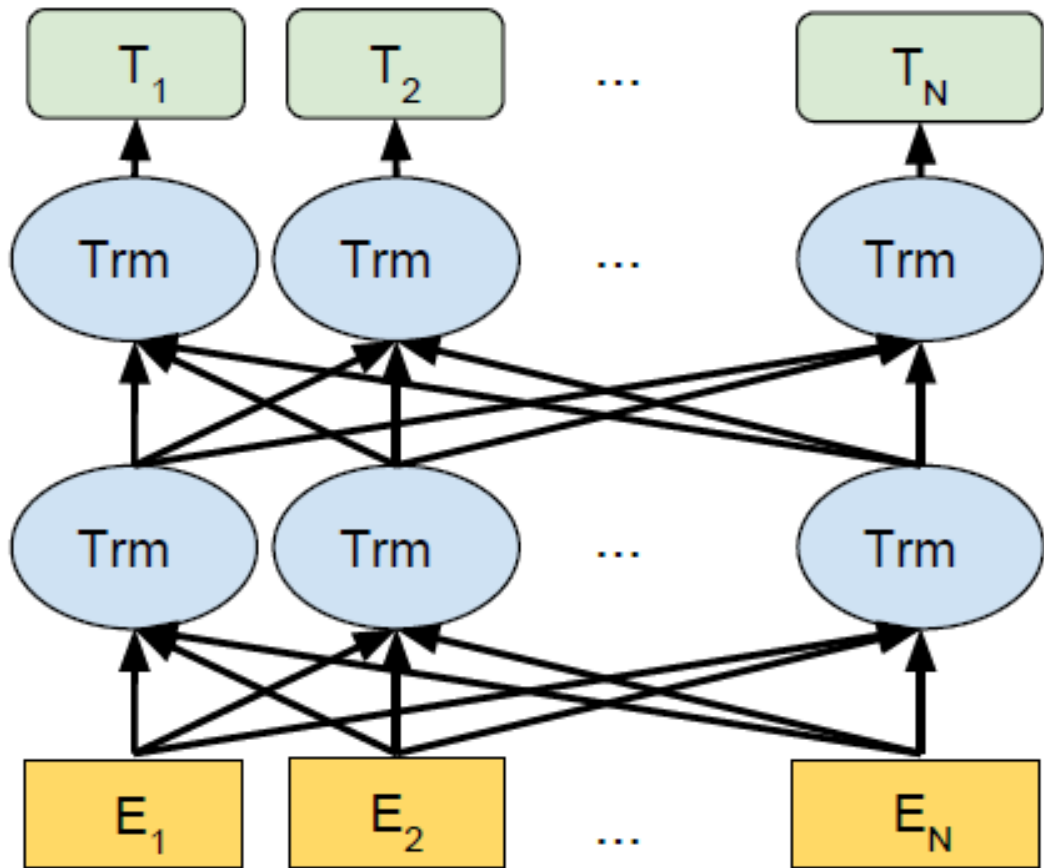


Figure 4 Inserting

Detail Methodology and calculations

start



A source \longrightarrow Select specific questions \longrightarrow Cleaning dataset.
Apply KNN Imputer.



Average score sentiment

End \longleftarrow Average DP Experience score at provider level.

Sentiment Generated



Step 1: Sourcing data from multiple teams and data preparation

Below data sources are collected from different platforms such as the cluster, migrated to our S3 storage, and collated together for further transformation.

[1] A source: Daily questions are designed to uncover directional insight to identify new issues, deep-dive known issues, quantified anecdotal feedback, and proactively identified anomalies.

[2] B source: The DP B source reviews are a outcome of the survey conducted by the teams. Drivers are given a free box where they can provide their feedback and the team tags the targeted topic of each feedback using word matching. These targeted topics are broad categories that drivers face today.

Step 2: How we calculate DP A source scores

Driver id	Q1	Q2	Q3
A	5		3
A	4.5	4	2
B	1		3
C		5	
C		3	2
D	3		1

* This table is for in A source.

Driver id	Q1	Q2	Q3
A	$(5+4.5)/2 = 4.75$	4	$(3+2)/2 = 2.5$
B	1		3
C		$(5+3)/2 = 4$	2
D	3		1

*Average the scores on DP level
Numbers are for indicative purpose only*

Driver id	Q1	Q2	Q3
-----------	----	----	----

Driver id	A source
A	3.75
B	2
C	3
D	2



A	4.75	4	2.5
B	1	2	3
C	3	4	2
D	3	2	1

**fill missing values
using KNN imputer*

Formula used for filling missing values

`knn = KNNImputer(n_neighbors=3,weights='distance')`

$\text{dist}(x,y) = \sqrt{\text{weight} * \text{sq. distance from present coordinates}}$ where, $\text{weight} = \frac{\text{Total \# of coordinates}}{\# \text{ of present coordinates}}$

Step 3: How we calculate B source scores

The feedback was given by the driver (I had a scary experience in the lot at the station with an aggressive fellow flex driver who bullied me and I felt pretty unsafe. I did not report him but just left as quickly as I could. I hope never to encounter him again.) goes into our BERT neural network model which generates sentiments between 1 to 5.

How BERT works:

BERT makes use of a Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task.

Driver id	feedback	Sentiment generated after applying BERT
A	Great program!	5

Step 4: - Overall Score

Driver id	feedback	Sentiment generated after applying BERT
A	Great flexible program!	5
B	Very poor experience	1
C	Not satisfied as much	2
D	Satisfied but need more attention	3
E	-----	FILLED WITH ENGAGEMENT SCORE

Driver id	A source
A	3.75
B	2
C	3
D	filled with engagement score

Driver id		Overall_score
A	$((B \text{ score} * \text{weight1}) + (A \text{ score} * \text{weight2})) / w1 + w2$	4.375
B	“	1.5
C	“	2.5
D	“	2.5
E	“	1

Plots and Transformations

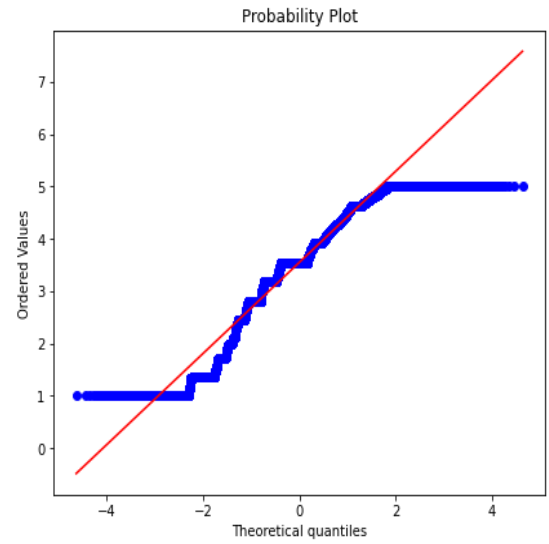
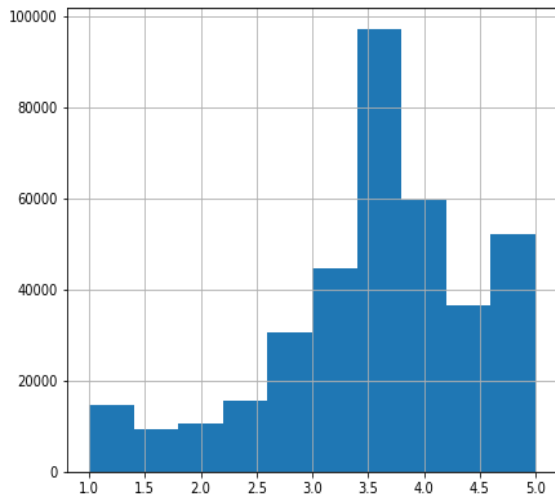


Figure 5 Overall score plot

DP Overall Experience plot and QQ-Plot

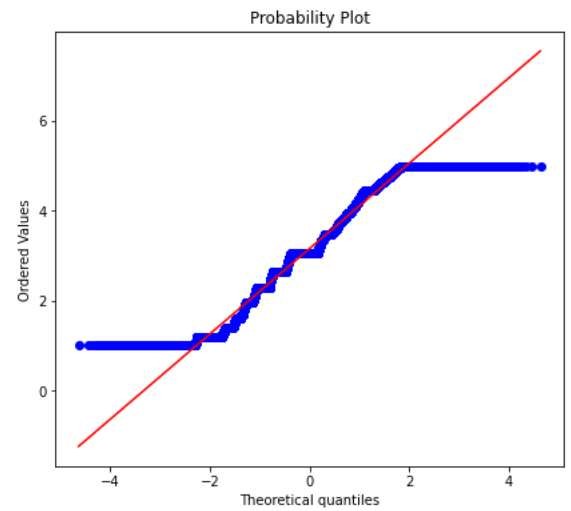
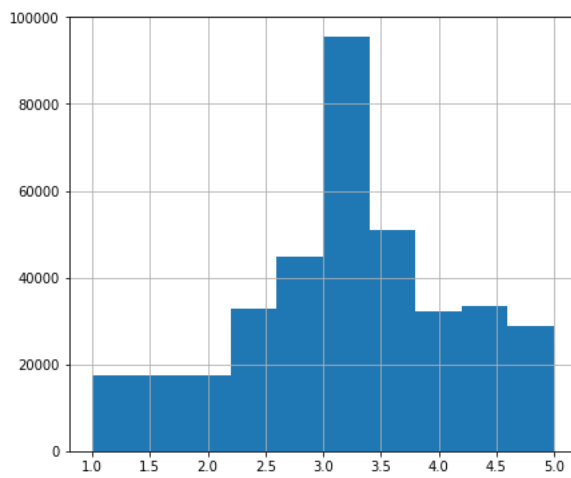


Figure 6 overall score plot 1

After applying Box-cox Transformation

Dummy Data

We created a dummy data and achieved these results on dummy data. We include two attributes in our data, A source and B source.

Provider			DP
ID	A	B	experience
1	34	18	3
2	9	19	4
3	38	8	1
4	13	17	2

SUMMARY OUTPUT

<i>Regression Statistics</i>	
Multiple R	0.18899026
R Square	0.03571732
Adjusted R Square	-0.0245503
Standard Error	1.46417452
Observations	35

ANOVA

	<i>df</i>	<i>SS</i>	<i>MS</i>	<i>F</i>	<i>Significance F</i>
Regression	2	2.54103207	1.27051603	0.59264477	0.55881734
Residual	32	68.6018251	2.14380703		
Total	34	71.1428571			

	<i>Coefficients</i>	<i>Standard Error</i>	<i>t Stat</i>	<i>P-value</i>	<i>Lower 95%</i>	<i>Upper 95%</i>
Intercept	2.32013944	0.83171962	2.7895692	0.00881726	0.62598201	
A	0.02307884	0.02465108	0.93622034	0.35617198	-0.0271338	-
B	-0.0021063	0.00892864	-0.2359083	0.81500726	-0.0202934	

1.4 Organization

Chapter I: Contains the Introduction, Problem statement, scope, Methodology of the System or Project.

Chapter 2: Discusses an abstract survey of the published papers and if any disadvantages are identified in the paper.

Chapter 3: Discusses the detailed requirement of the problem identified for the major project, system architecture and implementation details.

Chapter 4: Discusses the Performance Analysis of the model.

Chapter 5: Concludes the Report, Discusses any Future Scope.

Next Step

2] Identifying the top n levers which impact the DP experience.

After successfully quantifying the DP experience score as the dependent variable, we have identified these metrics. These are not exhaustive. We are working on identifying more.

Controllable:

Un-Controllable:

In order to perform the analysis, we will consider the granularity at a provider-id level for a duration of 01-01-2021 to 28-01-2022. After post data creation we will do bivariate analysis for the purpose of determining empirical relationship between the variables. Basis the bivariate results, we will do transformations to make data better fit. We will plot these data points and build correlation matrix to remove multicollinearity.

we will do regression analysis to find how an independent variable x influences y(DrEX). We will perform feature selection.

For better explanation on next steps, we ran a dummy model to explain the expected output. After that, we will run two separates machine learning models one is linear regression and the other is an artificial neural network model and compare the accuracy of both the models, which gave us better results in terms of accuracy.

In regression we will look at p-value and adjusted R squared value to determine the metric and model fit. We will use the coefficient output of linear regression ($y = ax_1 + bx_2 + \dots + K$) to determine the impact of independent variable on DrEX.

CHAPTER NO. 2

2 LITERATURE SURVEY

2.1 Survey

Arthur Samuel who first coined the term ML in 1959. According to him, ML is a “**Field of study that gives computers the ability to learn without being explicitly programmed**”.

Machine learning is the computer method of analysing data in various fields, to get the accuracy and future predictions of the data and many more requirements and knowledgeable information using different automated built models. There are many already inbuilt machine learning models present that focus on data analytics, data modelling, identifying and exploiting links against databases. Machine learning involves interactions between changes in the predictions that are generated considering the individual feature effects of the provided database. The various algorithms in machine learning algorithms that are being used are kNN(k- nearest neighbour Classifier), GMM(Gaussian mixture model), Naive Bayes Classifier, Bayes theorem, Random Forest, SVM, Boosting algorithm, EDA(Exploratory Data Analysis).

Machine learning is an important aspect of business as well as research.

It uses different algorithms and neural network models algorithms and neural networks to help the systems processing and progressing.

We found a research paper which is written by *Tianqi Chen* (University of Washington) on xgboost, which is a tree based boosting system.

2.2 Existing Applications

Business, Accounting, fashion, management, research and many more fields have machine learning based numerous applications. During the pandemic many applications based on time series algorithms were developed. Retailers, banks use ML models to keep track of the information of users.

Some existing applications are:-

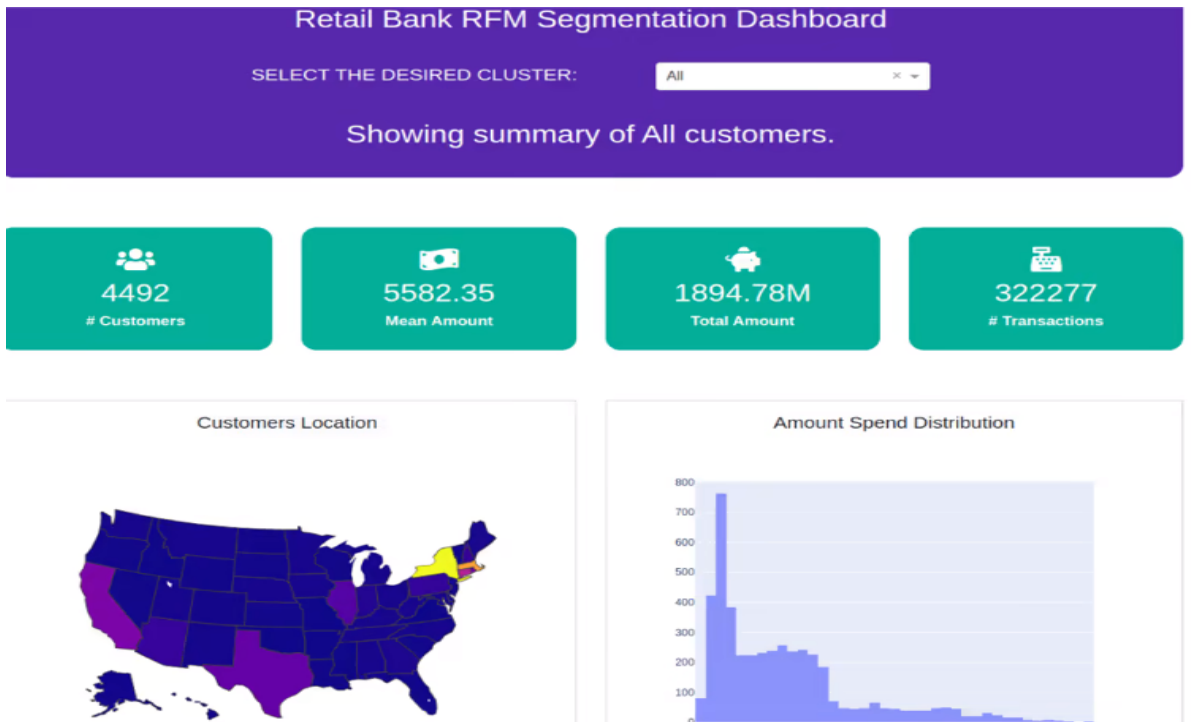


Figure 7 retail bank rfM

2.2.1 XGboost

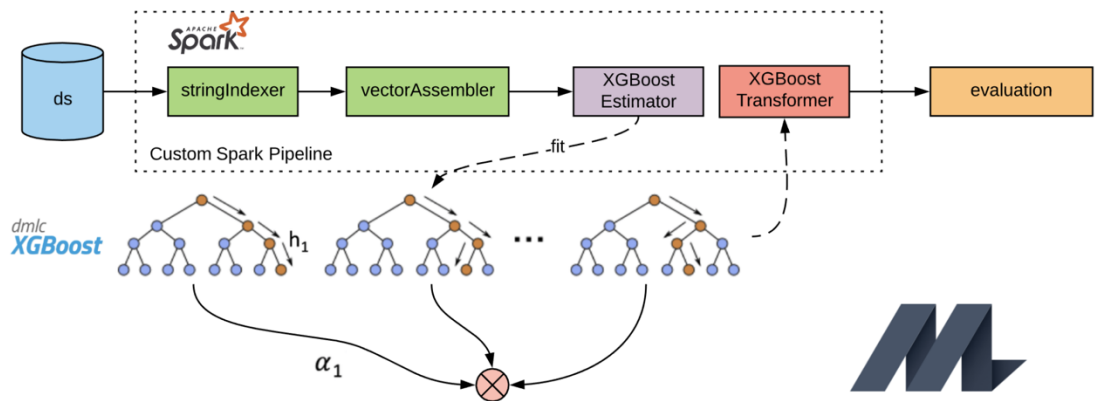


Figure 8 Xgboost

2.3 Proposed Application

The proposed Streamlit web application is a python based machine learning application that provides various functionalities like data analytics and experience predictions of delivery partner. This includes various python libraries that are being required for algorithms.

Machine learning algorithms available in this application for data analysis:-

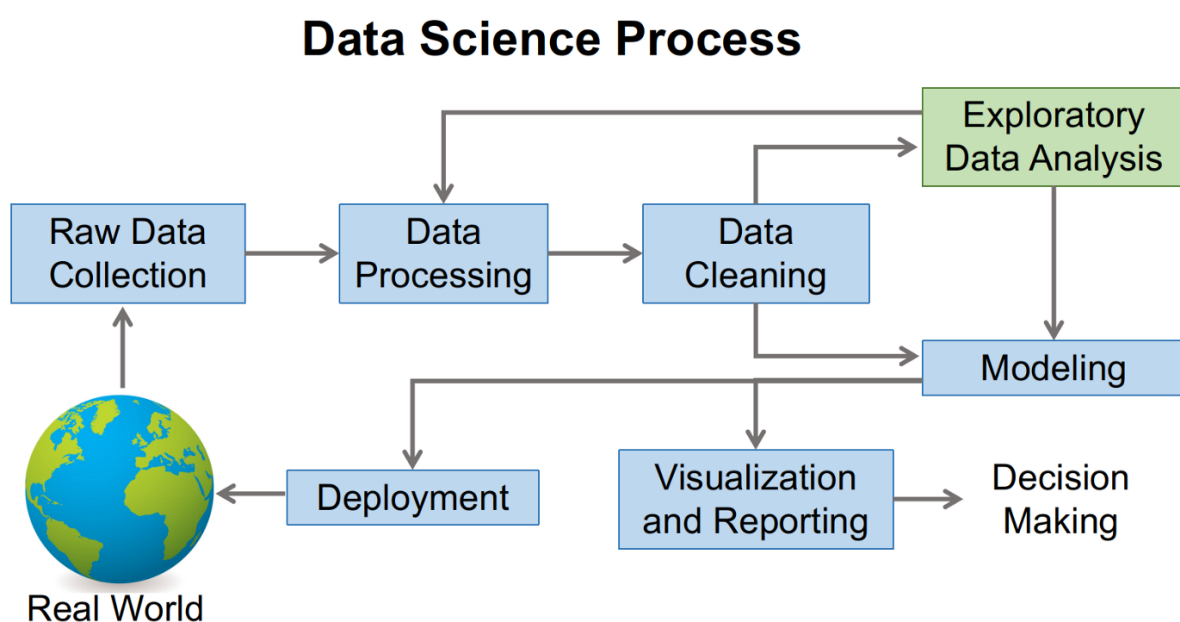


Figure 9 DS MODEL

2.4 Feasibility study

Key steps required for any software development are acquired. Allows the user to use applications that have been tested. A proper research has been carried out before development of application.

2.4.1 Economic feasibility

The designed web application does not require any investment during its formation. However we can make it profitable by applying subscription charges. This profit can be earned without any investment using this webapp.

2.4.2 Probability feasibility:-

The performance of the application and its accuracy of predicting the correct outcomes is high. The application works admirably during testing yet some malfunctioning may occur in rare cases. It entails the technical as well as researching knowledge.

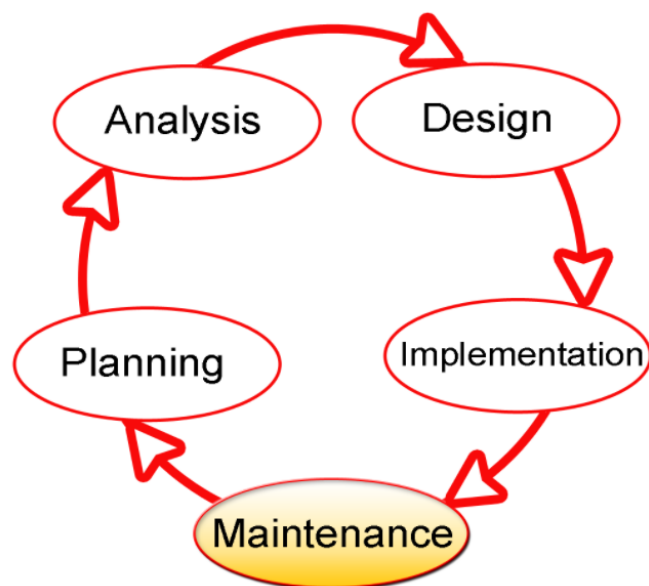
2.4.3 Technical feasibility

The developer is familiar with the application and the application is user friendly. Technical performance refers to the support of applications created with the current existing technology.

This application is feasible with android, laptop, computer and devices having internet connection and application installation system. Only an internet connection and device is needed to run the application. So, it is technically feasible.

CHAPTER NO. 3

3 SYSTEM DEVELOPMENT



3.1 Tools & Technologies Used

3.1.1 Python



It is a programming language with a wide range of capabilities. Its broad features make working with targeted programs (including meta-programming and meta-objects) simple, and it fully supports object-oriented programming. Python takes advantage of power typing as well as the integration of reference computation and waste management waste collecting. It also supports advanced word processing (late binding), which binds the way the words change during the process.

Patches to less essential sections of C Python that can give a minor improvement in performance at an obvious price are rejected by Python developers who try to prevent premature execution. When speed is crucial, the Python program developer can use mod-written modules in C-languages or PyPy, a timely compiler, to submit time-sensitive jobs. Python's architecture supports Lisp culture in terms of functionality. Filters, maps, and job reduction, as well as a list comprehension, dictionaries, sets, and generator expressions, are all included.

Two modules (itertools and functools) in the standard library use realistic Haskell and Standard ML tools.

3.1.2 Html



HTML is a markup language used to specify the format of a document that will be displayed on a computer screen. HTML pages may contain audio, moving graphics, lists, genuine data, and java documents and can be generated as basic text or

sophisticated multimedia. Web browsers, programs that may go across the network and show a range of information, display HTML pages. HTML is the most widely used web publication format. Allows the author to insert not just text but also text titles, lists, and tables, as well as still pictures, video, and audio in the text. Details can be retrieved from the student's computer. HTML sites may also be used to enter data and as a business transaction's front end.

3.1.3 CSS



CSS (Cascading Style Sheets) is not any technical programming language it is just used for styling the web page. Whereas HTML (Hyper Text Markup Language) is used to structure your webpage. CSS is widely used to design the web pages.

3.1.4 STREAMLIT



Streamlit is used to create the web application or web API'S. Streamlit is a very powerful framework which is mostly used in the field of ML and data science. By streamlit you can create the web app in hours not in a week. Streamlit offers many components and by using those components you can boost your work.

components:- Radio buttons, Checkboxes, Expander etc.



Monitoring apps

Tools that display trends and real-time insights.



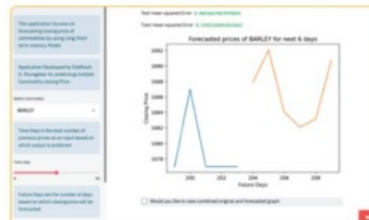
Analysis tools

Apps that use machine learning to analyze new data types.



Detection apps

Tools that use computer vision to detect and track objects.



Prediction tools

Apps that predict prices, stock shortages, quality issues, etc.



Explanatory apps

Apps that analyze large datasets and present easy to understand insights.



Interactive tools

Tools that allow you to interact with the data to gain new insights.

3.1.5 Scikit-Learn



Scikit -learn is perhaps Python’s most helpful machine learning package. The learning library includes several useful machine learning and mathematical modeling techniques, such as division, slowing, merging, and size reduction.

It’s one of Scikit-most Learn's popular APIs. The Estimator API is used with all of Scikit-machine Learn’s learning algorithms because it provides a uniform interface and a wide range of ML applications. The scale is the object read by the data (which contains data).

TensorFlow is a more advanced version of the standard library. Scikit-Learn is a cutting edge package that allows you to create learning algorithms for a variety of machines thus you may construct an object in one or a few lines of code and use it to

measure an Iron point or forecast a result.

3.1.6 Language Used in This Project

LANGUAGE USED

Table 3.1 Languages

S.NO	LANGUAGE USED
1.	PYTHON
2.	HTML/CSS

ALL LIBRARIES AND FRAMEWORK RELATED TO PYTHON

Table 3.2 Libraries

S.NO	PYTHON LIBRARY/Framework
1.	STREAMIT==1.2.0
2.	DTALE==1.61.1
3.	ALTAIR==4.1.0
4.	AUTOVIZ==0.0.81
5.	GITPYTHON==3.1.14
6.	FUTURE==0.18.2
7.	IPYTHON==7.21.0
8.	MATPLOTLIB==3.3.4
9.	NUMPY==1.20.1
10.	PANDAS==1.2.2
11.	PILLOW==8.4.0
12.	PLOTLY==4.14.3
13.	PYPARSING==2.4.7
14.	PYPERCLIP==1.8.2
15.	SCIKIT-LEARN==0.24.2
16.	SCIPY==1.6.1
17.	SEABORN==0.11.2
18.	SKLEARN==0.0
19.	STATSMODELS==0.12.2
20.	STRSIMPY==0.2.1

3.2 Requirements for Hardware and Software

- Intel Core i3, 5, i7. and 2 GHz processor RAM must be at 512MB.

- Hard disc with a capacity of at least 10 GB
- Input Keyboard and Mouse are the devices that are used.
- Monitor or PC as an output device
- Versions of Windows 7, 10, and above are supported.
- VS code as code editor.
- Python is the programming language.
- Streamlit cloud hosting for deployment.

3.3 Functional-Requirements

- The functional requirements for this model will be:
 - Select the operation you want to perform.
 - Upload the required csv file (if needed)
 - Do Prediction or Visualization

To Run:

Laptop or a Desktop

Internet Connection

For Development:

Intel Core i3 or above

4 GB RAM DDR4

Visual studio code

Jupyter Notebook

3.4 Non-Functional Requirements

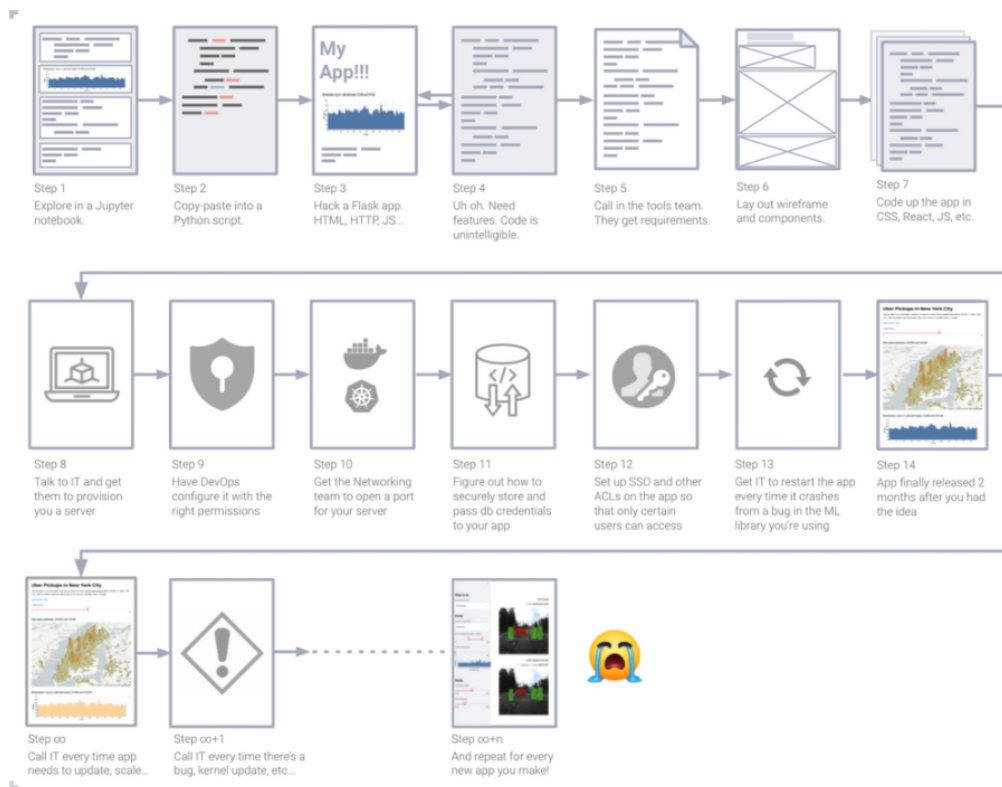
Table 3.3 NFR

Scalability	✓
Reliability	✓
Availability	✓
Recoverability	✓

3.5 Deployment of the model

Why Streamlit Cloud

Because in other hostings, there are many steps to do the deployment or they have to pay for the hosting.



But streamlit cloud hosting is much better because in just few clicks, one can deploy it easier and securely share, collaborate with the team.

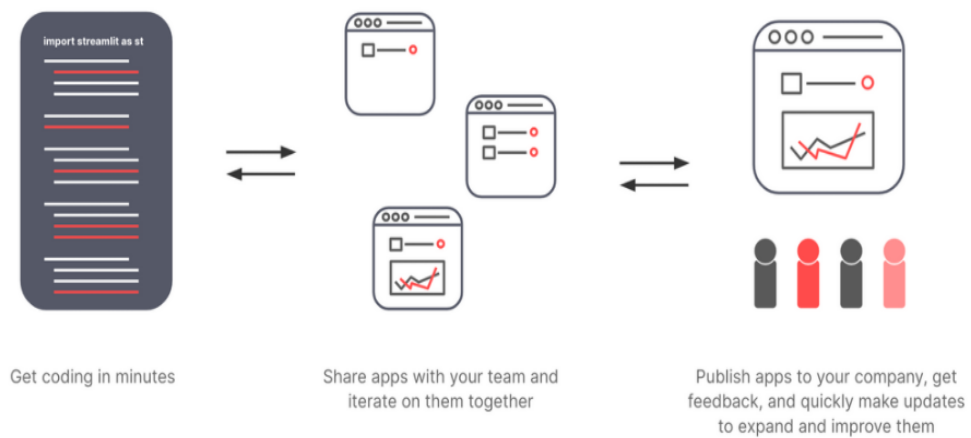


Figure 10 streamlit

1) Build and Deploy app

Build the streamlit app in your favorite IDE like VS code and commit the code and necessary files on github

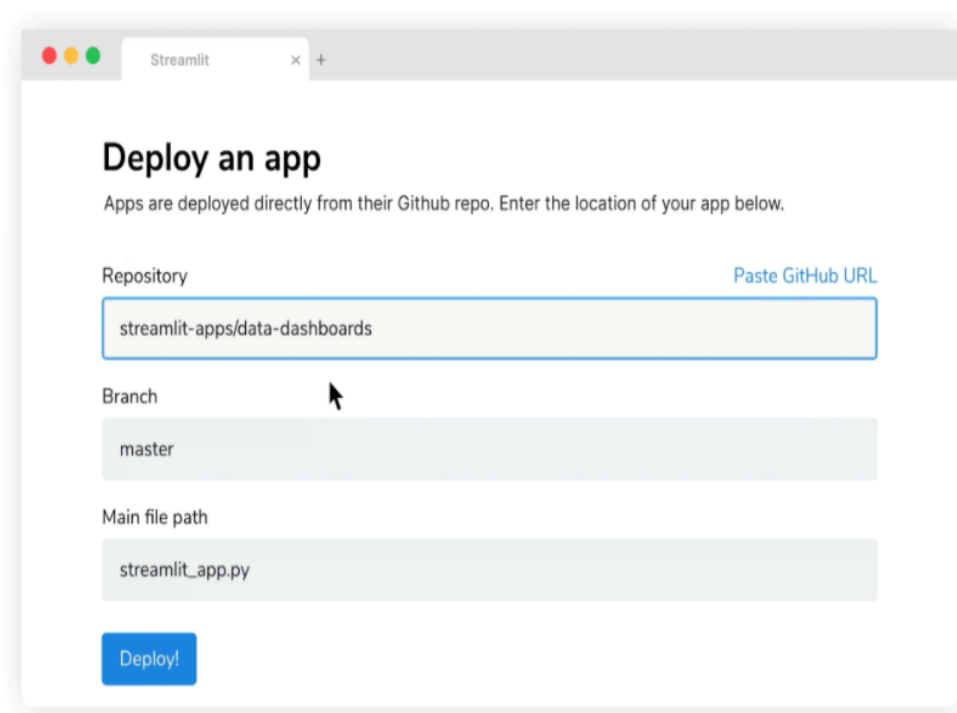


Figure 11 Deployment

2) Share your app securely

After successful deployment, one can easily share it with the whole team. Now, they can start using the web app directly. Also you can lock down the app so only particular people can view it.

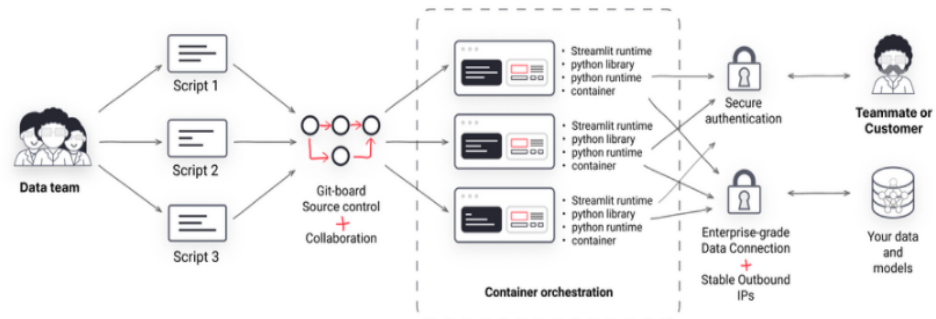


Figure 12 sharing of app

3) Iterate rapidly

As we have uploaded it on GitHub, so, now one can quickly iterate on it. This streamlit cloud hosting deploys the app continuously from GitHub, and gives the power of version-controlled development.

3.6 User Interface Functionalities Requirements

User interaction specification (UI specification) is a document that describes user usage. There is usually some basic software that requires a UI when it comes to case management and case management. The goal of improving usage conditions is for the UI designer to have a better knowledge of product features.

3.7 Flow Chart

1. Start
2. Choose the operation which you want to perform
3. Choose file upload/Choose parameter prediction
4. Predictions appear on the screen
5. Output end

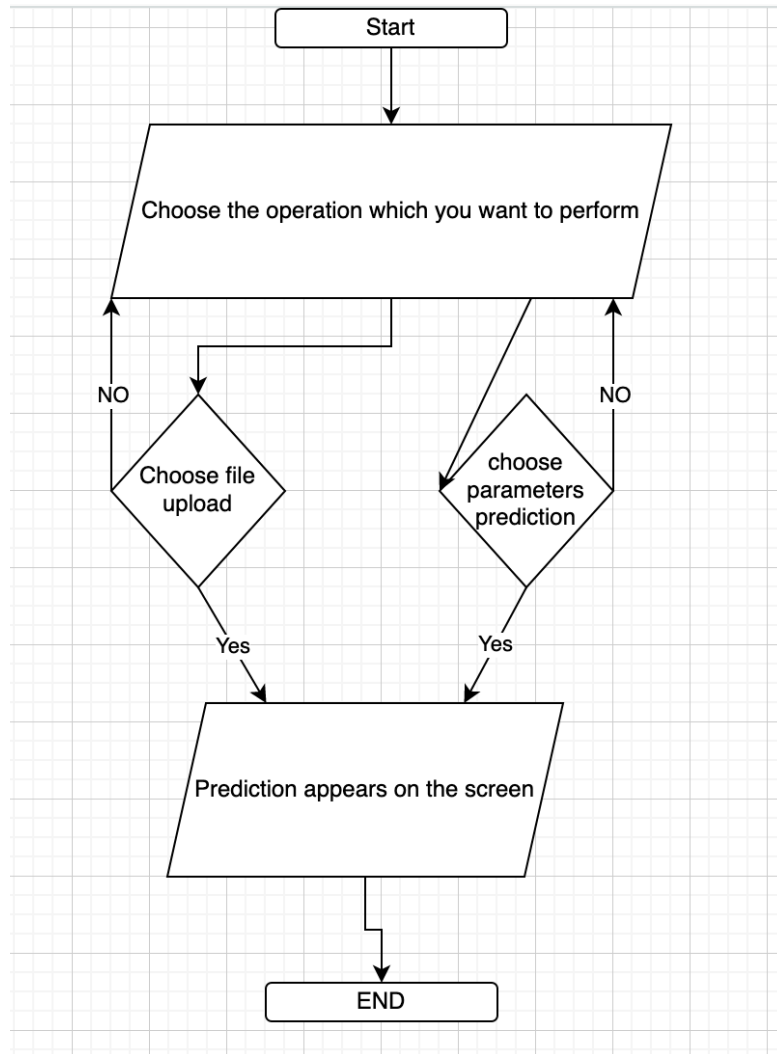


Figure 13 flow chart

3.8 Use Case Diagram

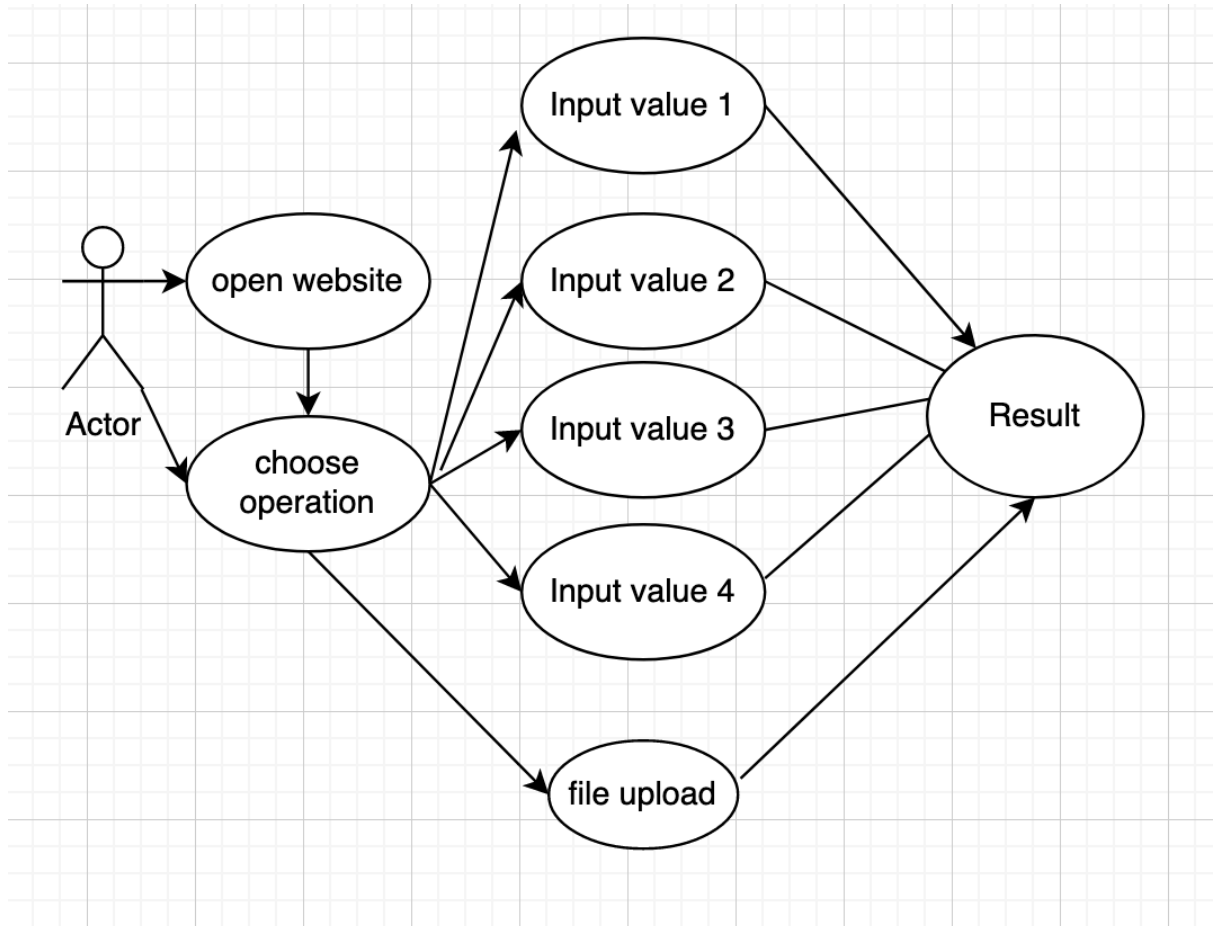


Figure 14 UC diagram

3.9 Machine Learning

As we know time series analysis and other traditional regression and classification models are different from each other. With the help of predictions a organization has the capacity to forecast the sales and result into many benefits.

- ➔ Improved customer satisfaction
- ➔ Working capital decreases
- ➔ Decreasing backlog of orders.

In this project XGBOOST model is used and as we know that xgboost is divided into 6 parts.

We know that xgboost is the extension of gradient boosting technique and combination of loss function and different regularization techniques.

- [1] Introduction
- [2] Xgboost objective function
- [3] Taylors theorem
- [4] Build next learners
- [5] Build classification with log loss optimization

In xgboost we need a base model in starting. In xgboost the probability set in starting is 0.5, regardless of whether you are using XG boost for regression or classification.

To understand the xgboost we took a example of drug classifier.

The default prediction Is that there is a 50% of change the drug is effective or not.

We can illustrate the initial prediction by adding a y-axis to our graph to represent the probability that the drug is effective and drawing a thick black line at 0.5 to represent a 50% chance that the drug is effective. We represent it with two green dots which represent the probability 1 and the two red dots which represents the drug is not effective. The residuals, the difference between the observed and predicted values show us how good the initial prediction is. Now just like in gradient boosting we fit an xgboost tree to the residuals.

However since we are using xgboost for classification we have a new formula for the similarity scores.

$$\text{Similarity} = \frac{\text{sum}(\text{residuals})^2}{\text{sum}(\text{previous probability}*(1-\text{previous probability})) + \text{lambda}}$$

The numerator represents the sum pf the residuals squared. In other words the numerator for classification is the same as the numerator for regression, and just like for regression the denominator contains lambda the regularization parameter. However the rest of the denominator is different. The denominator is just sum of the each observation of the previously predicted probability times 1 mines the previous predicted probability. Now lets build tree just like for regression each tree starts out as single leaf..and all of the residuals go to the leaf, now calculator similarity scores for the leaf and plug all the residuals in the leaf node, because we do not square the residuals before we add them together, they will cancel each other out..

Similarly calculate similarity scores for each leaf by splitting all the nodes.

Now calculate the gain for the each split:

$$\text{Gain} = \text{left}(\text{similarity}) + \text{right}(\text{similarity}) - \text{Root}(\text{similarity})$$

That's how overall Xgboost works to predict the experience of a driver by split into a different decisions trees.

So in this project for each driver we can predict the experience of a driver and also tells which factor influences the most driver experience out of 16 other attributes which we found by writing sql queries.



CHAPTER NO. 4

4 PERFORMANCE ANALYSIS

4.1 DS USED IN THE PROJECT

The data which is build by scratch using different sql queries used in this Project.

4.2 DATA SET FEATURES

4.2.1 Types of Data Set

Build from scartch	Numerical Dataset/Categorical dataset
--------------------	---------------------------------------

4.2.2 Number of Attributes, fields, and the data set description

Data Set	No. of Attributes	Description
Build from scratch	19	207000*19 (207000 rows and 19 columns)

4.3 Performance evaluation

For the performance evaluation we ran a model. We use automated python library pycaret which compares more than 15 models at a time and we got a result in regression mentioned below but the results we got is not satisfying so we turn our regression model into classification model to obtain the results:-

1. Regression model output:-

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
rf	Random Forest Regressor	1.9092	5.5639	2.3587	0.0946	0.2816	0.2826	2.3000
catboost	CatBoost Regressor	1.9346	5.6001	2.3664	0.0887	0.2857	0.2922	7.7990
xgboost	Extreme Gradient Boosting	1.9285	5.6074	2.3679	0.0875	0.2853	0.2905	0.6230
lightgbm	Light Gradient Boosting Machine	1.9414	5.6150	2.3695	0.0863	0.2863	0.2936	0.0790
gbr	Gradient Boosting Regressor	1.9568	5.6778	2.3827	0.0761	0.2878	0.2962	0.9080
et	Extra Trees Regressor	1.9225	5.8130	2.4109	0.0540	0.2866	0.2838	1.4040
br	Bayesian Ridge	2.0319	6.0420	2.4580	0.0168	0.2952	0.3075	0.0230
lr	Linear Regression	2.0321	6.0433	2.4582	0.0166	0.2952	0.3075	0.0190
ridge	Ridge Regression	2.0321	6.0433	2.4582	0.0166	0.2952	0.3075	0.0180
lar	Least Angle Regression	2.0321	6.0433	2.4582	0.0166	0.2952	0.3075	0.0180
en	Elastic Net	2.0365	6.0611	2.4618	0.0137	0.2959	0.3086	0.0180
lasso	Lasso Regression	2.0405	6.0845	2.4666	0.0099	0.2964	0.3092	0.0180
omp	Orthogonal Matching Pursuit	2.0403	6.0963	2.4690	0.0080	0.2966	0.3092	0.0170
llar	Lasso Least Angle Regression	2.0579	6.1471	2.4792	-0.0002	0.2976	0.3116	0.0190
ada	AdaBoost Regressor	2.0834	6.1806	2.4859	-0.0056	0.2900	0.2928	0.1890
knn	K Neighbors Regressor	2.1081	6.8162	2.6107	-0.1093	0.3064	0.3109	0.4060
huber	Huber Regressor	2.0503	7.1566	2.6571	-0.1651	0.3016	0.3196	0.1510
par	Passive Aggressive Regressor	2.5070	10.4771	3.1857	-0.7084	0.3703	0.3642	0.0350
dt	Decision Tree Regressor	2.4186	10.6725	3.2667	-0.7368	0.3939	0.3392	0.0680

Figure 15 regression output

2. Classification model output

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.6384	0.7239	0.4836	0.6269	0.6305	0.3154	0.3173	8.8360
xgboost	Extreme Gradient Boosting	0.6204	0.6887	0.4287	0.5909	0.5975	0.2553	0.2612	14.6970
et	Extra Trees Classifier	0.6180	0.7093	0.4645	0.6107	0.6119	0.2823	0.2843	6.8410
catboost	CatBoost Classifier	0.6167	0.6856	0.4255	0.5867	0.5930	0.2469	0.2535	44.0560
lightgbm	Light Gradient Boosting Machine	0.6162	0.6834	0.4234	0.5835	0.5925	0.2460	0.2517	3.5090
gbc	Gradient Boosting Classifier	0.5893	0.6739	0.4247	0.5786	0.5836	0.2300	0.2302	22.3160
dt	Decision Tree Classifier	0.5574	0.6068	0.4505	0.5713	0.5637	0.2035	0.2038	2.9280
ada	Ada Boost Classifier	0.5183	0.6480	0.4259	0.5747	0.5424	0.1807	0.1838	4.1910
knn	K Neighbors Classifier	0.4436	0.5850	0.3987	0.5318	0.4782	0.1022	0.1068	6.8660
ridge	Ridge Classifier	0.3649	0.0000	0.3783	0.4918	0.3639	0.0470	0.0576	2.3780
lda	Linear Discriminant Analysis	0.3646	0.5660	0.3773	0.4849	0.3811	0.0445	0.0525	2.5180
lr	Logistic Regression	0.3560	0.5868	0.4009	0.4934	0.3767	0.0643	0.0787	7.3010
svm	SVM - Linear Kernel	0.3055	0.0000	0.3595	0.5412	0.3015	0.0451	0.0622	4.2420
qda	Quadratic Discriminant Analysis	0.1271	0.5619	0.3545	0.5280	0.1382	0.0128	0.0304	2.5080
nb	Naive Bayes	0.1268	0.5618	0.3544	0.5269	0.1377	0.0124	0.0295	2.5040

Figure 16 classification output

3. Auc curve of our model

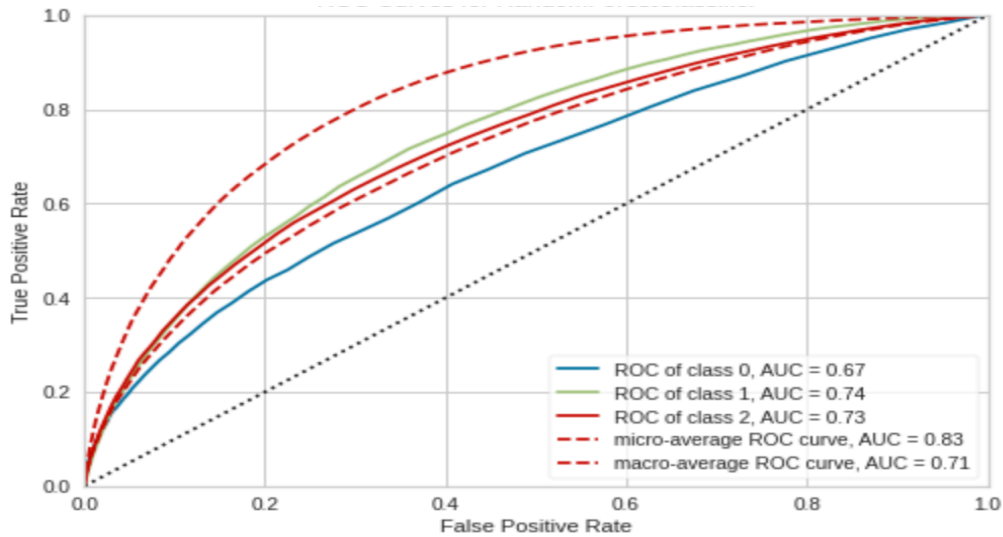
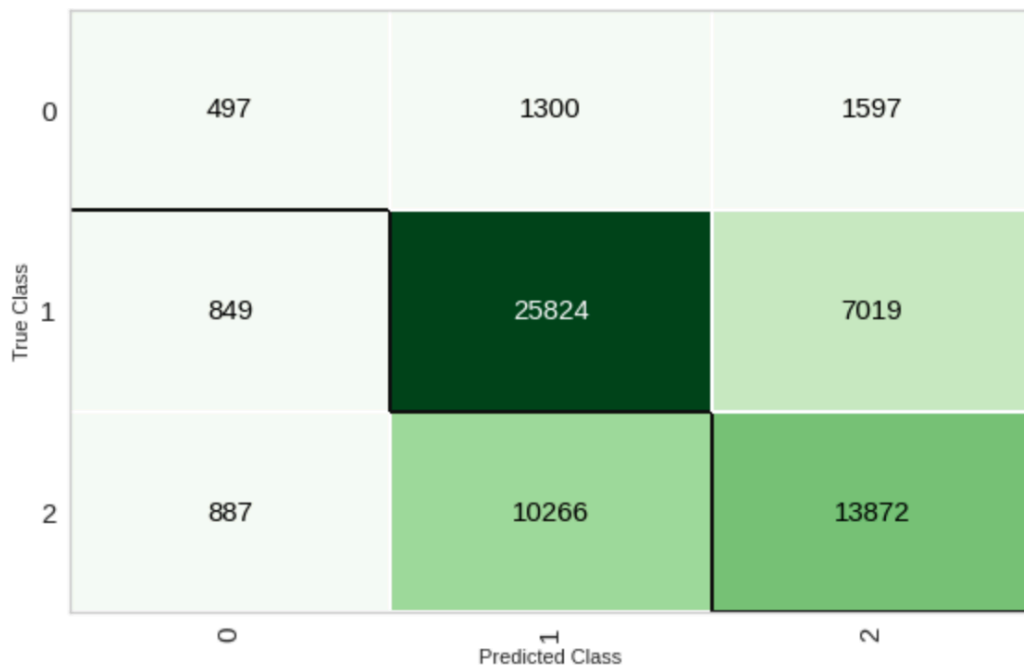
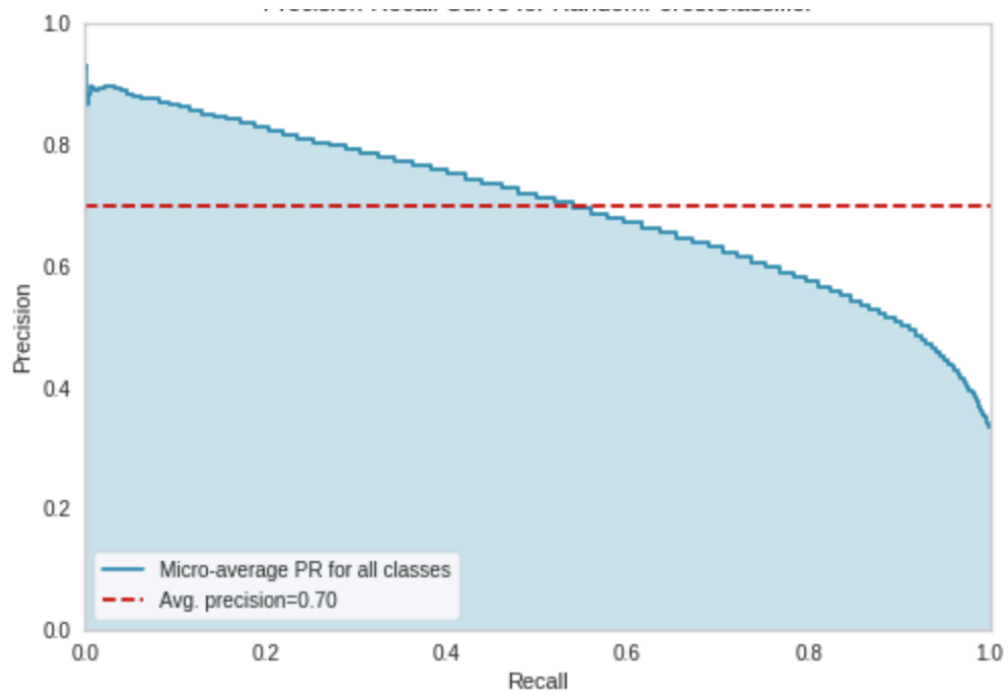


Figure 17 AUC

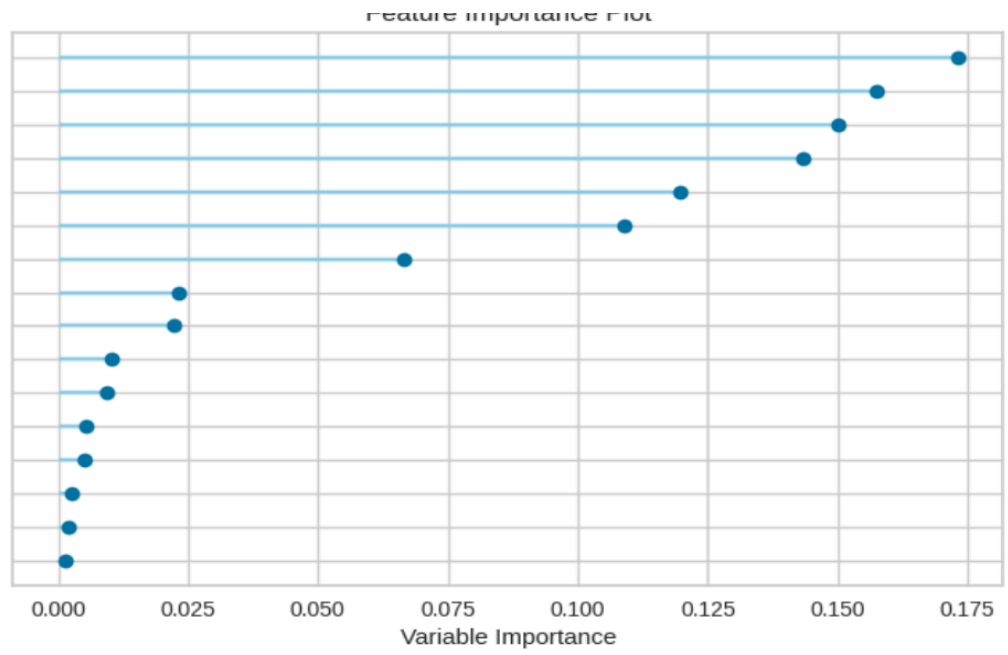
4. Confusion matrix of our model



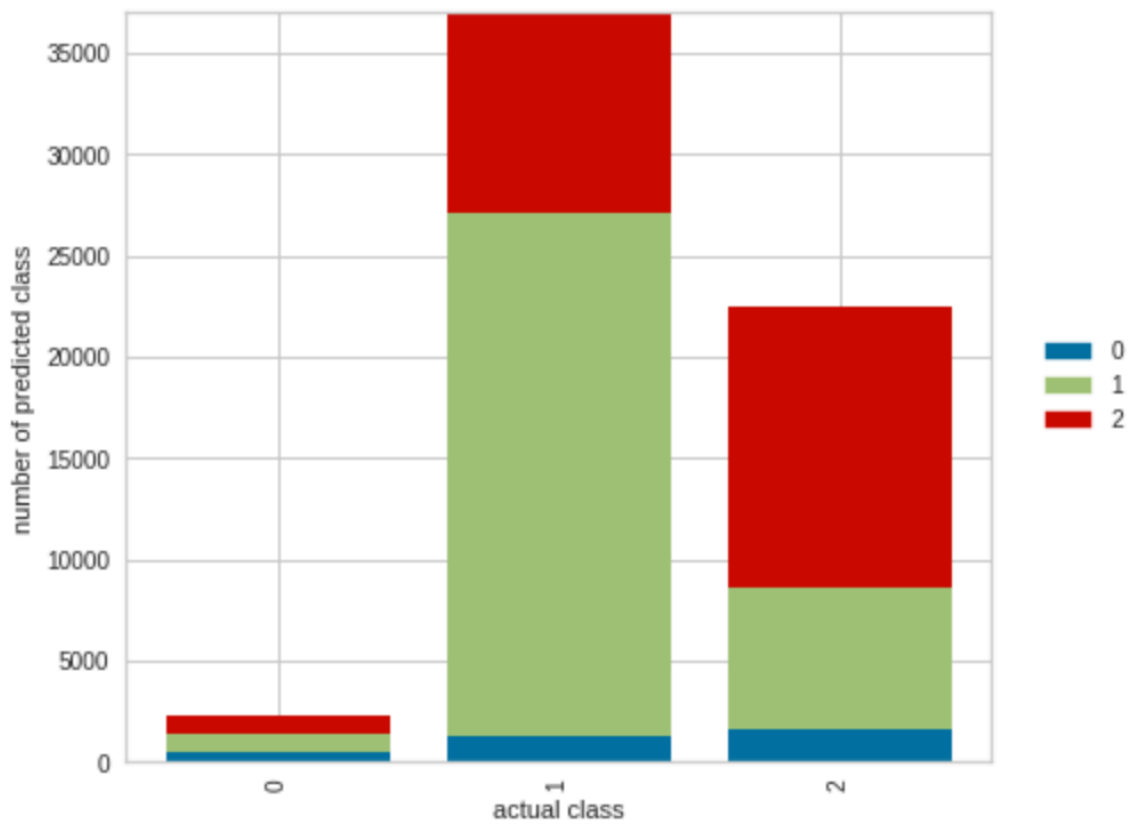
5. Precision and recall of our model



6. Variable importance of our model



7. Prediction error of our model



4.4 Various Results and Output At Different Stages and Website UI:

This gives us few instructions on how to use the model, allows us to select desired option

- Batch Upload
- Online upload
- Excel to interactive dashboard

Online upload

- This selected option allows users to perform prediction for the delivery partner by input the parameters.
- Helping them to become knowledgeable and productive.

Batch upload

- This is about analysing the data to gain insights and be productive.
- It gives an elaborated analytic report.

Overview of all dashboards and results we are getting

Sheet 1 NPS Overview

Net promoter score is calculated based on only one question in A source (Would you recommend delivering with our company to a friend or relative?) It is a widely used market research metric that typically takes the form of a single survey question asking respondents to rate the likelihood that they would recommend company to a friend or relative.

Sheet 2 CSAT Overview

DP's Satisfaction score is calculated based on only one question in A source (Overall, how satisfied are you delivering packages with our company?
CSAT is a short for customer satisfaction which is commonly used key performance indicator used to track how satisfied customers are with company.

Sheet 3 CET Overview

It is a metric derived from a customer satisfaction survey that measures a product or services ease of use to customers. Customer effort score reflects the amount of effort of a customer had to exert to use a product or service. CET is calculated based on a particular question how easy was to pick up a package from a station.

DP Net promoter score

It is an indicator of the future growth of the company on the basis of dp's loyalty. Net promoter score is a widely used market research metric that typically takes the form of a single survey question asking respondents to rate the likelihood that they would recommend a company, product, or a service to a friend or colleague.

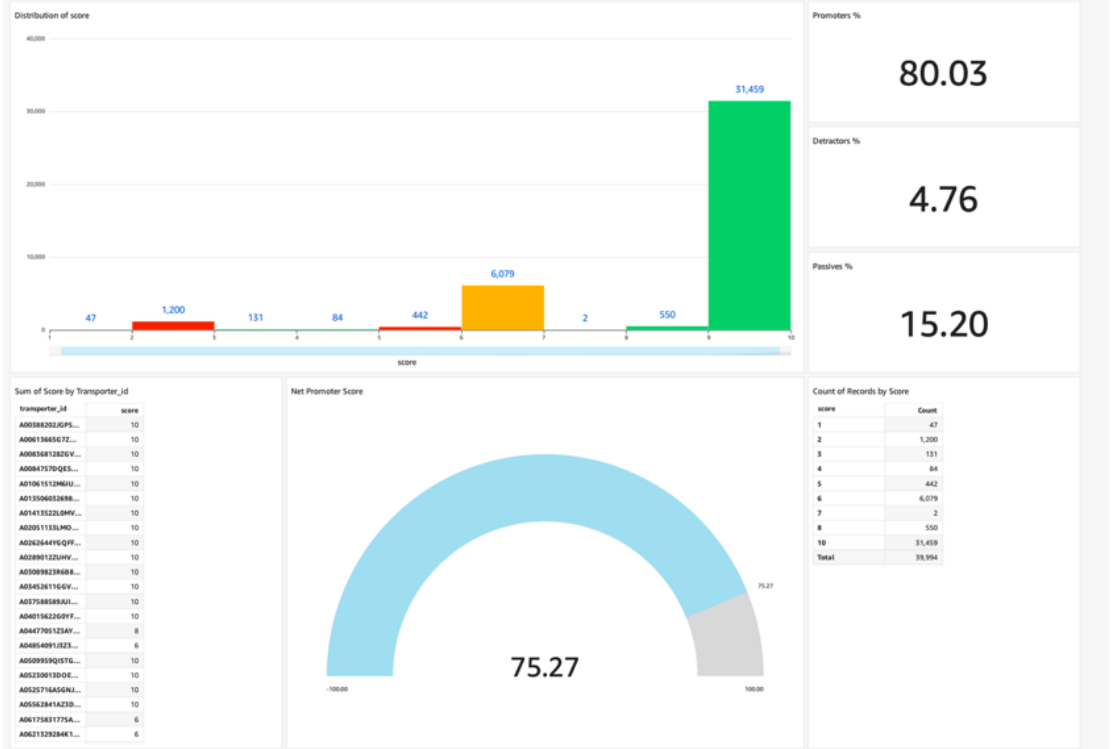
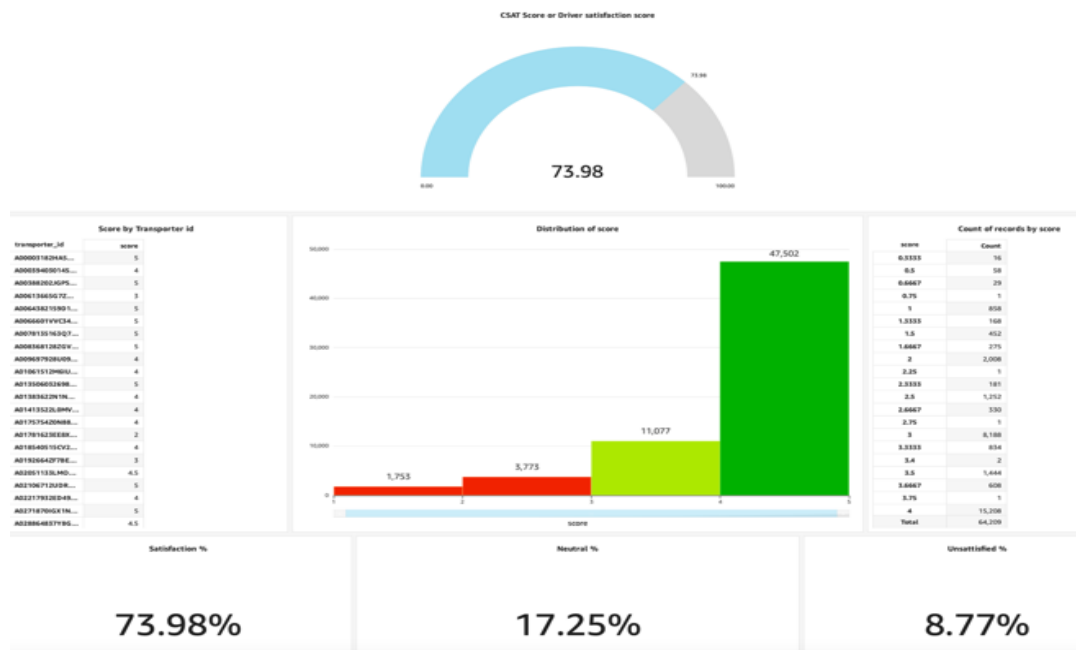
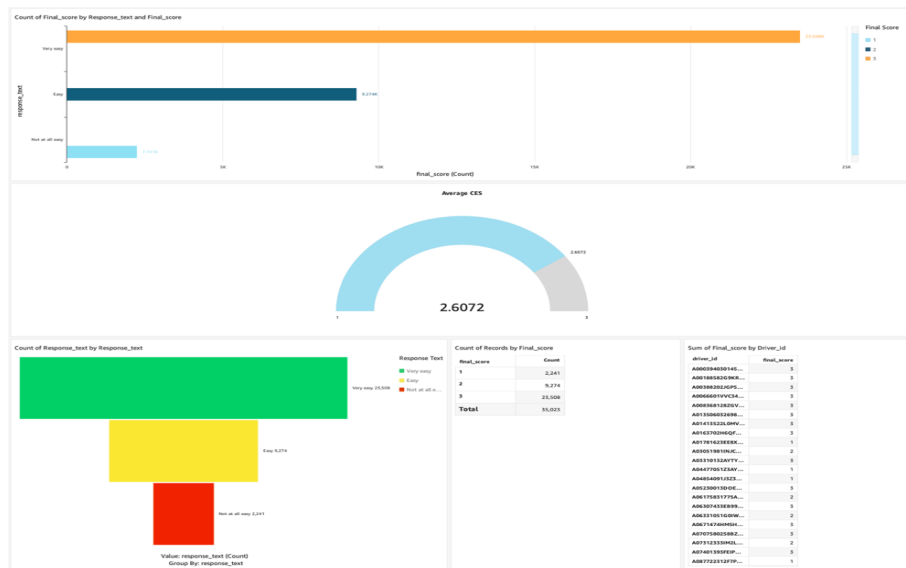
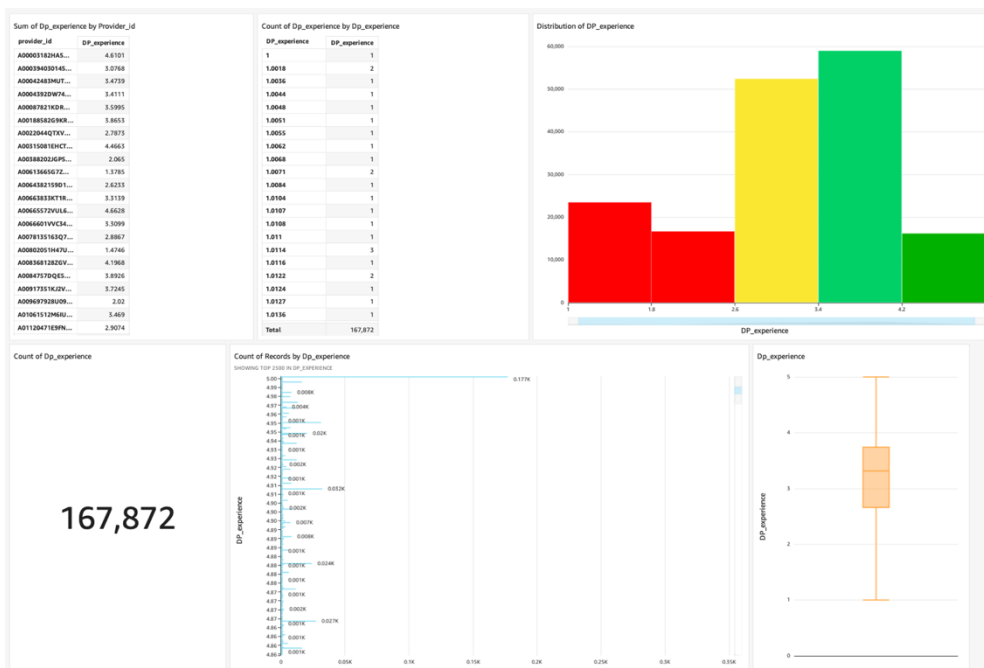


Figure 18 Dashboard





Sheet 4 Results we are getting after quantify overall DP experience score
 This Dashboard created on dummy data



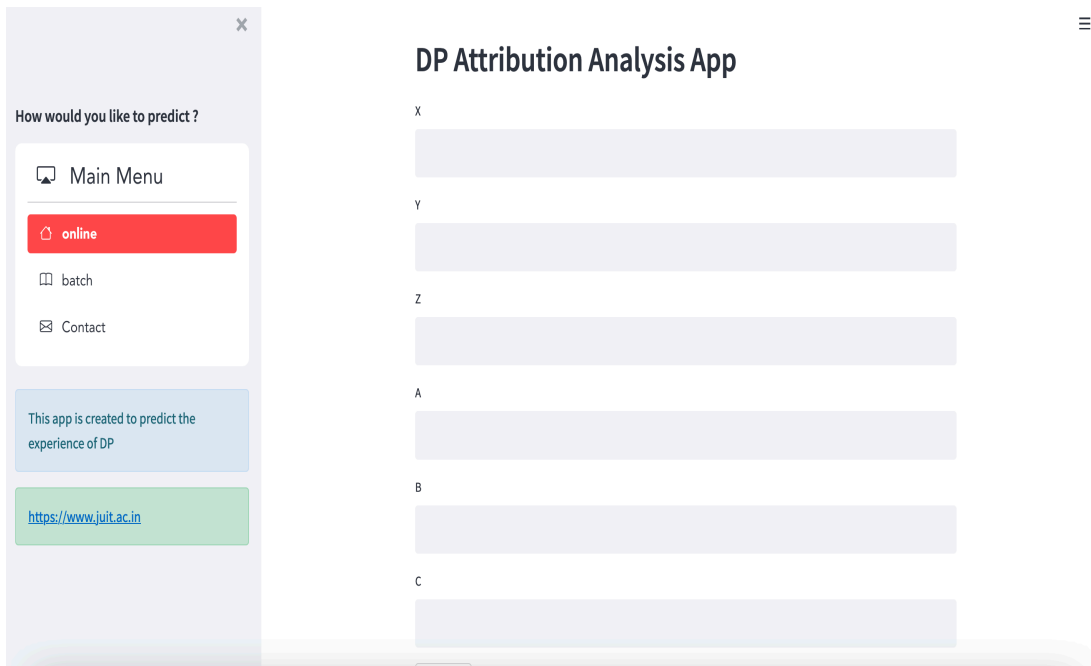
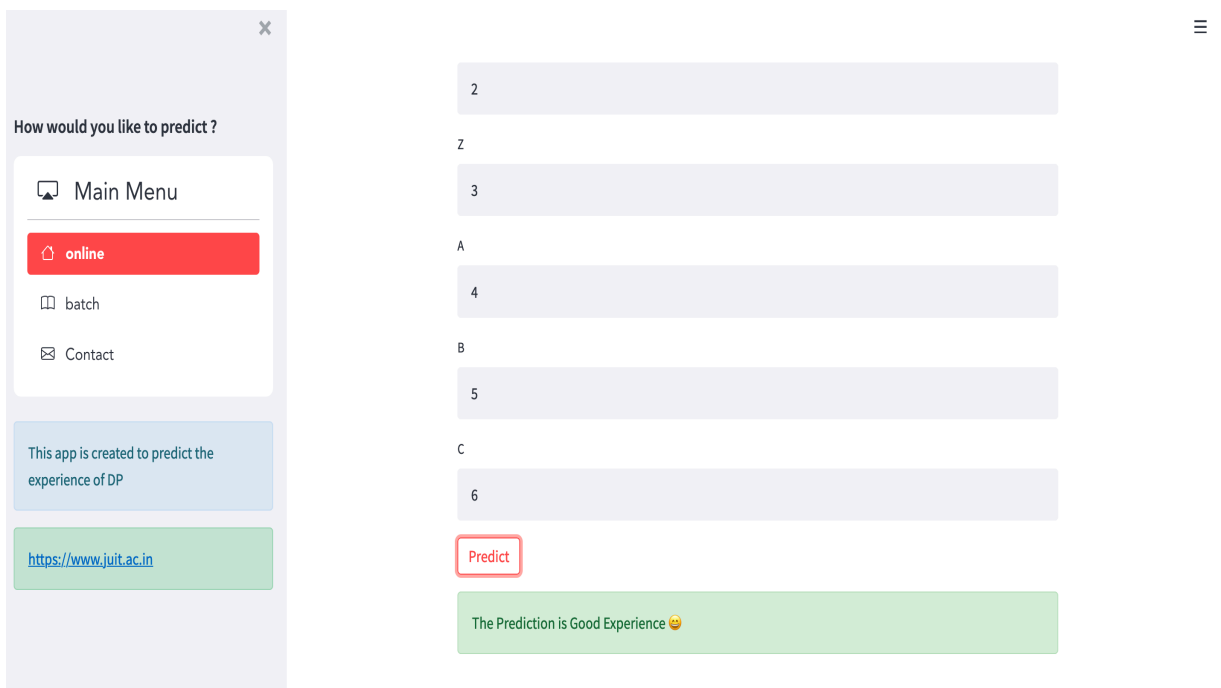


Figure 19 user interface



How would you like to predict ?

Main Menu

- online
- batch
- Contact

This app is created to predict the experience of DP

<https://www.juit.ac.in>

Y
0

Z
0

A
0

B
0

C
0

Predict

The Prediction is Neutral Experience 😊

How would you like to predict ?

Main Menu

- online
- batch
- Contact

This app is created to predict the experience of DP

<https://www.juit.ac.in>

Y
0.63

Z
11.04

A
23.27

B
35.3

C
1

Predict

The Prediction is Bad Experience 😞

☰ Main Menu

🏠 online

📁 batch

✉ Contact

This app is created to predict the experience of DP

<https://www.juit.ac.in>

DP Attribution Analysis App

Upload csv file for predictions



Drag and drop file here

Limit 200MB per file • CSV

Browse files

Made with Streamlit

🏃 RUNNING... Stop ☰

How would you like to predict ?

☰ Main Menu

🏠 online

📁 batch

✉ Contact

This app is created to predict the experience of DP

<https://www.juit.ac.in>



DP Attribution Analysis App

Upload csv file for predictions



Drag and drop file here

Limit 200MB per file • CSV

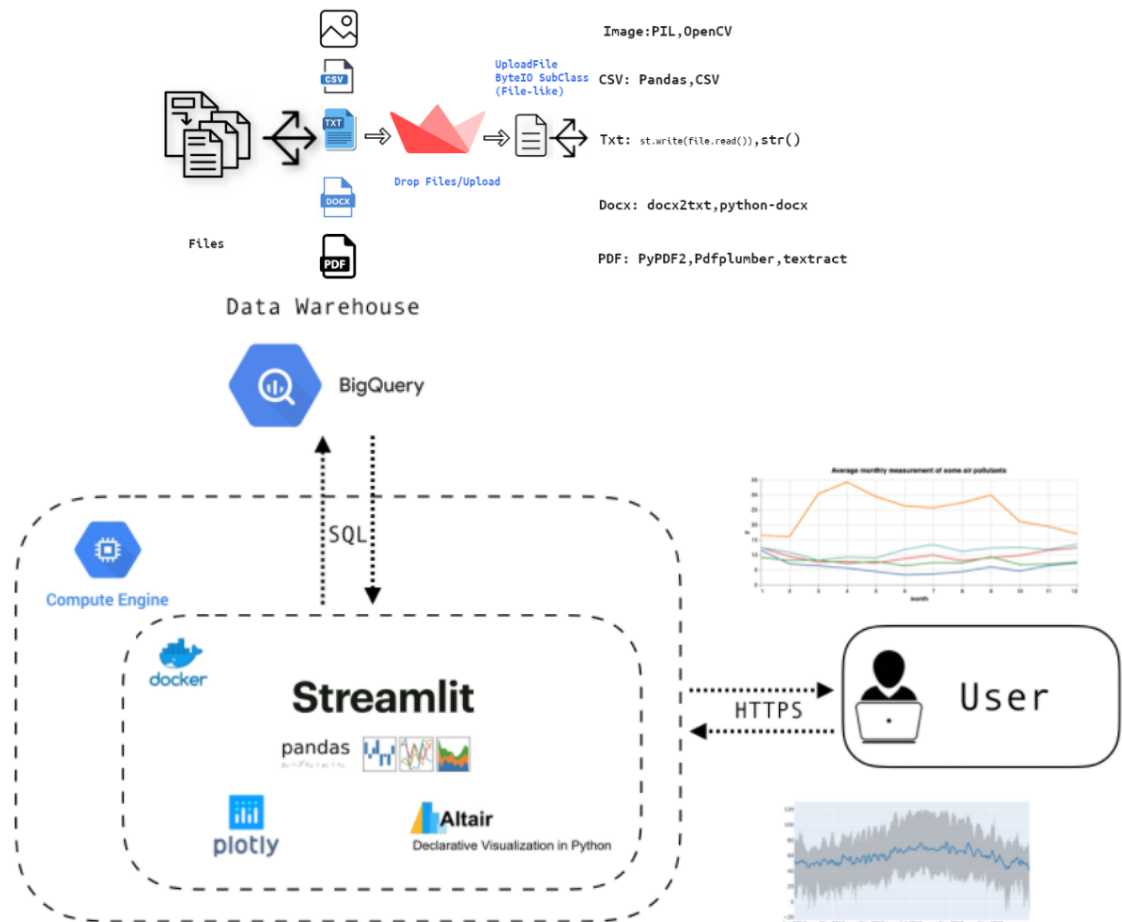
Browse files



model.csv 1.6MB

X

File Upload with Streamlit: A Simple Guide



CHAPTER NO. 5

5 Conclusion

5.1 Conclusion

Data Science how it can be implemented and used to create various applications to assist us and guide us to choose efficient solution.

The advantages of this web application are, it is simple to use gives accurate data, helps predicting the customer experience which really helps to analyze and take business decisions efficiently.

We use xgboost in our model and got around 65 percent of accuracy all details will be available in performance evaluation.

It also helps to analyze the performance and take right decisions.

The disadvantages are the file sharing function is under development,

Needs to improvise on implementations of algorithms, need to be more user friendly and elaborated so it can be useful to newbies, and have to make interpretations much easier.

Since the purpose of the thesis is to understand and implementation of Data science Algorithms in this application is most important and we believe that problems could be solved to make it much more suitable.

After testing more than 15 models we use Xgboost only to identify the prediction.

We identified that the accuracy is 65% in xgboost.

Learnings and Challenges

The biggest challenge we faced in the project was that it involved in-depth knowledge of business as well techniques to execute it. I invested in learning the DP Experience (DPX) space by reaching out to space owners and leveraging reading material available through Wiki and documents. I quickly ramped up on DPX space and identified all the touchpoints of DP through which they express their experience. As we moved further, as we faced challenges in including all of these factors in the analysis due to data constraints, availability and restricted access. In order to make the data processing more efficient, I learnt and leveraged pyspark over pandas. In order to unblock myself from providing visual insights, I learnt Quick sight on the job to create the dashboards.

Once I had the data-framework ready, in order to identify the right environment to aggregate data and run models I went through the SDS bootcamp where I learnt to set up my cloud desktop and used it to connect to S3. As this was an advanced analytics project, I spent hours learning and researching about the statistical techniques to make out model robust. Through this journey, I learnt about the nuances of descriptive and inferential statistics. I researched on NLP, BERT Neural network, Regression modelling, Hypothesis testing, Chi-Sq test, t-test etc. in order to identify the right of methodology with high accuracy.

5.2 Future Scope

Currently our web app is only able to take data and perform various Data science Functions but there is a functionality that we want it to have in future is to be able to share the results among the group of people in which consumer interacts and are important to consumer in product assessment.

6 REFERENCES

[1] Introduction to Taylor's theorem:

https://mathinsight.org/taylors_theorem_multivariable_introduction

[2] *Tianqi Chen, Carlos Guestrin*: XGBoost: A Scalable Tree Boosting System

<https://arxiv.org/abs/1603.02754>

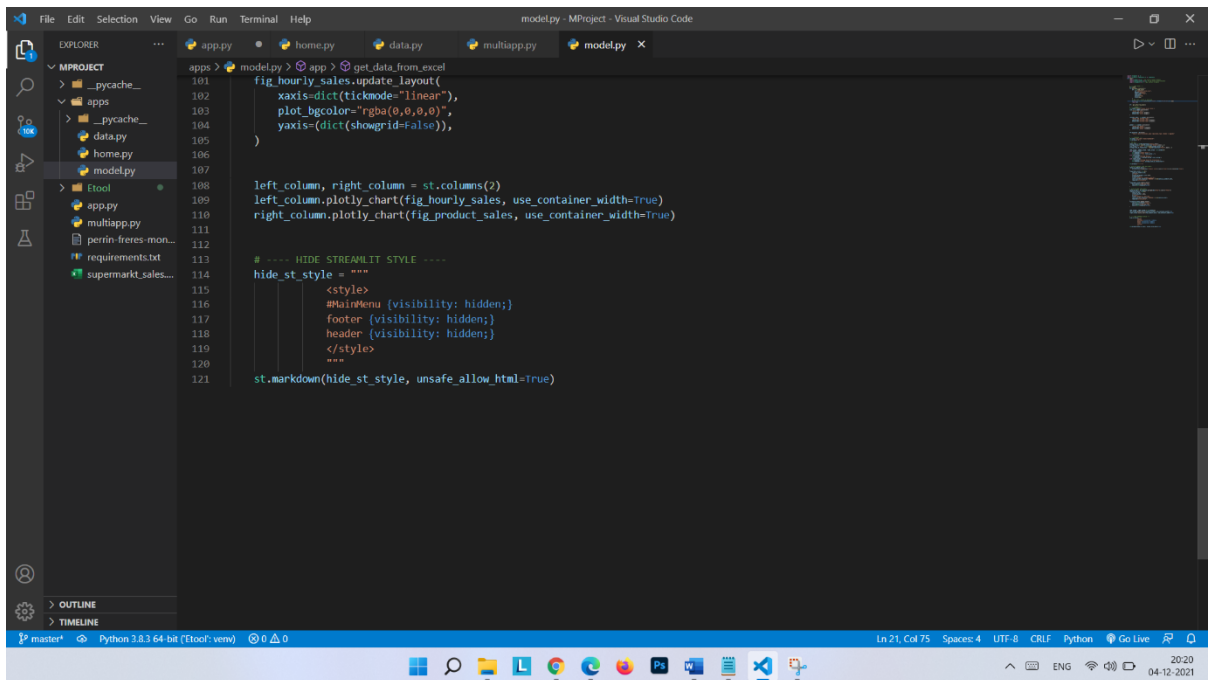
[3] *Tianqi Chen*: Introduction to Boosted Trees:

<https://homes.cs.washington.edu/~tqchen/pdf/BoostedTree.pdf>

[4] How to calculate gradient and hessian of log loss objective function:

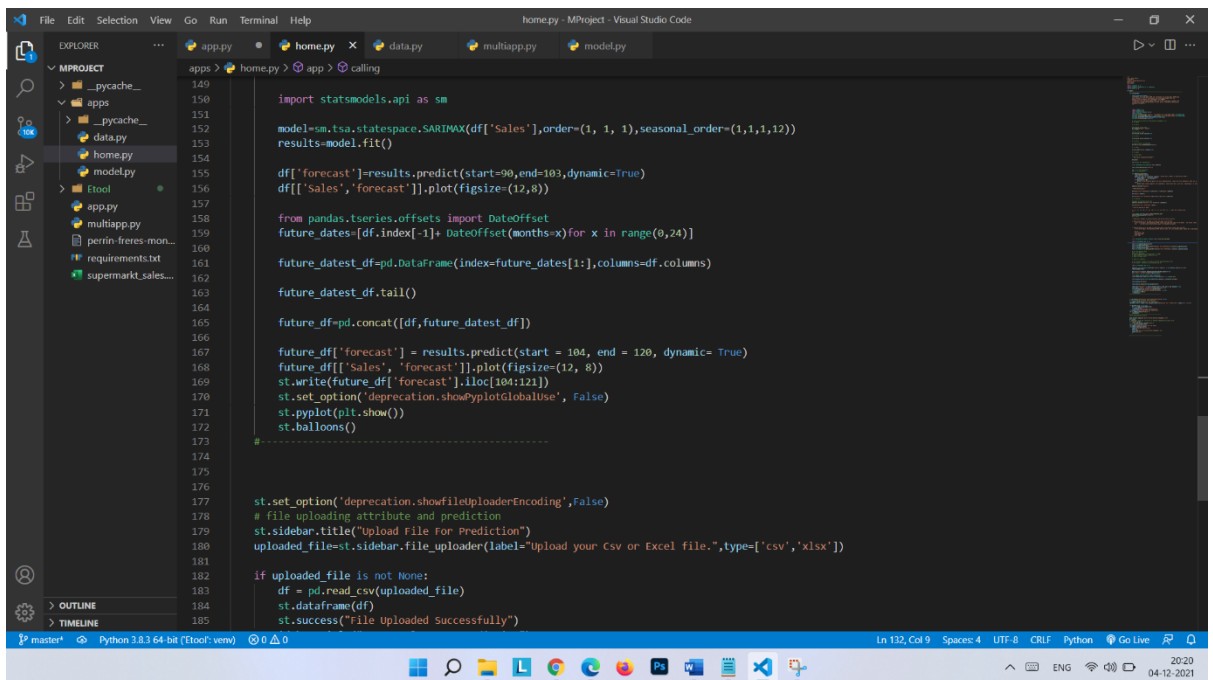
<https://stats.stackexchange.com/questions/231220/how-to-compute-the-gradient-and-hessian-of-logarithmic-loss-question-is-based>

7 APPENDICES



```
101 def get_data_from_excel:
102     fig_hourly_sales.update_layout(
103         xaxis=dict(tickmode="linear"),
104         plot_bgcolor="rgb(0,0,0)",
105         yaxis=(dict(showgrid=False)),
106     )
107
108 left_column, right_column = st.columns(2)
109 left_column.plotly_chart(fig_hourly_sales, use_container_width=True)
110 right_column.plotly_chart(fig_product_sales, use_container_width=True)
111
112 # --- HIDE STREAMLIT STYLE ---
113 hide_st_style = """
114     <style>
115     #MainMenu {visibility: hidden;}
116     footer {visibility: hidden;}
117     header {visibility: hidden;}
118     </style>
119     """
120
121 st.markdown(hide_st_style, unsafe_allow_html=True)
```

Figure 7.1 Code Snippets



```
149
150
151 import statsmodels.api as sm
152
153 model=sm.tsa.statespace.SARIMAX(df[\"Sales\"],order=(1, 1, 1),seasonal_order=(1,1,1,12))
154 results=model.fit()
155
156 df[\"forecast\"]=results.predict(start=90,end=103,dynamic=True)
157 df[[\"Sales\", \"forecast\"]].plot(figsize=(12,8))
158
159 from pandas.tseries.offsets import DateOffset
160 future_dates=[df.index[-1]+ DateOffset(months=x) for x in range(0,24)]
161
162 future_datest_df=pd.DataFrame(index=future_dates[1:],columns=df.columns)
163
164 future_datest_df.tail()
165
166 future_df=pd.concat([df,future_datest_df])
167
168 future_df[\"forecast\"] = results.predict(start = 104, end = 120, dynamic = True)
169 future_df[[\"Sales\", \"forecast\"]].plot(figsize=(12, 8))
170 st.write(future_df[\"forecast\"].iloc[104:121])
171 st.set_option('deprecation.showPyplotGlobalUse', False)
172 st.pyplot(plt.show())
173
174 #-----
175
176
177 st.set_option('deprecation.showfileuploaderEncoding', False)
178 # file uploading attribute and prediction
179 st.sidebar.title(\"Upload File for Prediction\")
180 uploaded_file=st.sidebar.file_uploader(label=\"Upload your Csv or Excel file.\",type=[\"csv\", \"xlsx\"] )
181
182 if uploaded_file is not None:
183     df = pd.read_csv(uploaded_file)
184     st.dataframe(df)
185     st.success(\"File Uploaded Successfully\")
```

```
File Edit Selection View Go Run Terminal Help model.py - MProject - Visual Studio Code
EXPLORER
MPROJECT
  _pycache_
  apps
  _pycache_
  data.py
  home.py
  model.py
  Etool
  app.py
  multiapp.py
  perin-freres-mon...
  requirements.txt
  supermarket_sales...
OUTLINE
TIMELINE
Python 3.8.3 64-bit (Etool: venv) 0 0
Ln 21, Col 75 Spaces: 4 UTF-8 CRLF Python Go Live
2020 04-12-2021
```

```
1 import streamlit as st
2 import streamlit.components.v1 as components
3 def app():
4     import pandas as pd # pip install pandas openpyxl
5     import plotly.express as px # pip install plotly-express
6     import streamlit as st # pip install streamlit
7
8
9
10 # ---- READ EXCEL ----
11 @st.cache
12 def get_data_from_excel():
13     df = pd.read_excel(
14         io="supermarkt_sales.xlsx",
15         engine="openpyxl",
16         sheet_name="Sales",
17         skiprows=3,
18         usecols="B:R",
19         nrows=1000,
20     )
21     # Add 'hour' column to dataframe
22     df["hour"] = pd.to_datetime(df["time"], format="%H:%M:%S").dt.hour
23     return df
24
25
26 df = get_data_from_excel()
27
28 # ---- SIDEBAR ----
29 st.sidebar.header("Please Filter Here:")
30 city = st.sidebar.multiselect(
31     "Select the City:",
32     options=df["City"].unique(),
33     default=df["City"].unique()
34 )
35
36 customer_type = st.sidebar.multiselect(
37     "Select the Customer type:",
38     options=df["customer_type"].unique(),
39     default=df["customer_type"].unique(),
40 )
```

```
File Edit Selection View Go Run Terminal Help multiapp.py - MProject - Visual Studio Code
EXPLORER
MPROJECT
  _pycache_
  apps
  _pycache_
  data.py
  home.py
  model.py
  Etool
  app.py
  multiapp.py
  perin-freres-mon...
  requirements.txt
  supermarket_sales...
OUTLINE
TIMELINE
Python 3.8.3 64-bit (Etool: venv) 0 0
Ln 11, Col 25 Spaces: 4 UTF-8 CRLF Python Go Live
2020 04-12-2021
```

```
1 """Frameworks for running multiple Streamlit applications as a single app.
2 """
3 import streamlit as st
4
5 class MultiApp:
6
7     def __init__(self):
8         self.apps = []
9
10    def add_app(self, title, func):
11        """Adds a new application.
12        Parameters
13        -----
14        func:
15            the python function to render this app.
16        title:
17            title of the app. Appears in the dropdown in the sidebar.
18        """
19        self.apps.append({
20            "title": title,
21            "function": func
22        })
23
24    def run(self):
25        # app = st.sidebar.radio(
26        app = st.selectbox(
27            'Select options below to navigate through the web app',
28            self.apps,
29            format_func=lambda app: app['title'])
30
31        app["function"]()
```

```
File Edit Selection View Go Run Terminal Help
datapy - MProject - Visual Studio Code

EXPLORER
MPROJECT
  _pycache_
  apps
  _pycache_
  datapy
  home.py
  model.py
  Etool
  app.py
  multiapp.py
  perrin-freres-mon...
  requirements.txt
  supermarket_sales...

OUTLINE
TIMELINE

Python 3.8.3 64-bit (Etool: venv)
Ln 24, Col 42 Spaces: 4 UTF-8 CRLF Python Go Live
20:20 04-12-2021
```

```
1 import streamlit as st
2 import streamlit.components.v1 as components
3 import pandas as pd
4
5 st.markdown("<link rel='stylesheet' href='https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css' integrity='sha384-Gn5384qqaI9g4Rs0mL9q410zqL1684tEcq36v96fPb6fyt4IwVtHh8E263Xmfc31SAw1GgFAw/dA1563Xm' crossorigin='anonymous'>", unsafe_allow_html=True)
6
7 def app():
8     def calling_Auto(df):
9         # import seaborn as sns
10        # df=sns.load_dataset('planets')
11        import dtale
12        from dtale.views import startup
13        from dtale.app import get_instance
14        startup(data_id="1", data=df)
15        df = get_instance("1").data
16        #components.html("<html><body><iframe src='/dtale/main/1' style='height:800px;width:800px;' /></body></html>")
17        st.markdown("<a href='/dtale/main/1' class='btn btn-outline-success' target='_blank'>Exploratory Data Analysis</a>", unsafe_allow_html=True)
18        components.html("<a href='/dtale/main/1' class='btn btn-primary' target='_blank'>Exploratory Data Analysis</a>")
19
20    st.set_option('deprecation.showFileUploaderEncoding',False)
21    st.sidebar.title("upload Any File To Do EDA")
22    uploaded_file=st.sidebar.file_uploader(label="Upload your csv or Excel file.",type=['csv','xlsx'])
23    if uploaded_file is not None:
24        df = pd.read_csv(uploaded_file) |
25        st.dataframe(df)
26
27
28    if st.sidebar.button("Start The Operation"):
29        calling_Auto(df)
```

```
File Edit Selection View Go Run Terminal Help
home.py - MProject - Visual Studio Code

EXPLORER
MPROJECT
  _pycache_
  apps
  _pycache_
  datapy
  home.py
  model.py
  Etool
  app.py
  multiapp.py
  perrin-freres-mon...
  requirements.txt
  supermarket_sales...

OUTLINE
TIMELINE

Python 3.8.3 64-bit (Etool: venv)
Ln 132, Col 9 Spaces: 4 UTF-8 CRLF Python Go Live
20:20 04-12-2021
```

```
1 """
2 Name- Rajat Gupta
3 Licensable
4 Contact- support@infojio.com
5 Batch-cs76
6 Roll-181429
7 """
8 import streamlit as st
9 import streamlit.components.v1 as components
10 import pandas as pd
11
12 def app():
13     #-----
14     def calling(df):
15
16         ARIMA_DEPRECATION_ERROR = """
17         statsmodels.tsa.arima_model.ARMA and statsmodels.tsa.arima_model.ARIMA have
18         been removed in favor of statsmodels.tsa.arima_model.ARIMA (note the .
19         between arima and model) and statsmodels.tsa.SARIMAX.
20         statsmodels.tsa.arima_model.ARIMA makes use of the statespace framework and
21         is both well tested and maintained. It also offers alternative specialized
22         parameter estimators.
23         """
24
25         import numpy as np
26         import pandas as pd
27         import matplotlib.pyplot as plt
28         import warnings
29         warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARMA',FutureWarning)
30         warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARIMA',FutureWarning)
31         warnings.warn(ARIMA_DEPRECATION_ERROR,FutureWarning)
32         warnings.filterwarnings("ignore")
33
34         # df=pd.read_csv('perrin-freres-monthly-champagne-.csv')
35         # df.head()
```

```
1 # Import all the necessary libraries
2 from matplotlib import pyplot
3 import pandas as pd
4 import streamlit as st
5 import streamlit.components.v1 as components
6 from multiapp import MultiApp
7 from apps import home, data, model # import your app modules here
8
9 # Include important bootstrap cdn meta library
10 st.markdown(<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1ao6KA
11 +999RXx06fy41W7th0E263Xmrc1LS4wJgkAw/dA1S63Xm" crossorigin="anonymous">, unsafe_allow_html=True)
12
13 # Navbar using bootstrap
14 st.markdown("""
15 <nav class="navbar fixed-top navbar-expand-lg navbar-dark" style="background-color: #3498db;">
16 <a class="navbar-brand" href="#" target="blank">Saas Tool</a>
17 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
18 aria-label="Toggle navigation">
19 <span class="navbar-toggler-icon"></span>
20 </button>
21 <div class="collapse navbar-collapse" id="navbarNav">
22 <ul class="navbar-nav">
23 <li class="nav-item active">
24 <a class="nav-link disabled" href="#">Home <span class="sr-only">(current)</span></a>
25 </li>
26 <li class="nav-item">
27 <a class="nav-link" href="#" target="blank">File Sharing</a>
28 </li>
29 </ul>
30 </div>
31 </nav>
32 """, unsafe_allow_html=True)
33
34
35 app = MultiApp()
```


PLAG REPORT

Rajat Gupta

ORIGINALITY REPORT

9 %	7 %	2 %	6 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	stackoverflow.com Internet Source	2 %
2	Submitted to Universitat Politècnica de València Student Paper	1 %
3	towardsdatascience.com Internet Source	1 %
4	www.Hotjar.Com Internet Source	1 %
5	github.com Internet Source	1 %
6	Submitted to University of Northumbria at Newcastle Student Paper	<1 %
7	Submitted to The University of the South Pacific Student Paper	<1 %
8	Submitted to Westford School of Management Student Paper	<1 %

9	Submitted to University College London Student Paper	<1 %
10	Submitted to CSU, San Diego State University Student Paper	<1 %
11	Submitted to Pinellas County Schools Student Paper	<1 %
12	quizlet.com Internet Source	<1 %
13	pypi.org Internet Source	<1 %
14	www.qualtrics.com Internet Source	<1 %
15	Www.qualtrics.com Internet Source	<1 %
16	dspace.carthage.edu Internet Source	<1 %
17	eprints.fri.uni-lj.si Internet Source	<1 %
18	acadpubl.eu Internet Source	<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches < 3 words