Project Report

GLOBAL SALES DATA ANALYTICS

TEAM ID : PNT2022TMID07570

TEAM MEMBERS:

S.SATHISH KUMAR

D.SRIMANJUNATH

B.SARAVANAN

J.SHYAM

K.NIRMAL RAJ

A.RAGUL RAM PRASHANTH

INDEX

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas
3.2 Ideation & Brainstorming
3.3 Proposed Solution
3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement
4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams
5.2 Solution & Technical Architecture
5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation
6.2 Sprint Delivery Schedule
6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1
7.2 Feature 2
7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases
8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

INTRODUCTION

- **Project Overview**

Global sales data analytics refers to the technology and processes used to gather sales data and gauge sales performance. Sales leaders use these metrics to set goals, improve internal processes, and forecast future sales and revenue more accurately.

sales analytics tool turn the data into an easy-to-digest format. As they say, a picture is worth a thousand words, so the more visual make the data, the better. the tool have features that reduce friction and make it simple for sales teams to focus on selling rather than create more work.

Generally, sales analytics is divided into four categories , On that in this project we have used the predictive analytics technique.

Predictive analytics is taking what you've learned about past sales and using it to gauge patterns and trends. This allows you to make educated predictions.

- **PURPOSE**

Sales analytics is the sales team's hidden superpower. It can enable your agents to spot key trends, dive deep, predict outcomes, and increase productivity.

Accurate analysis also gives the team the ability to tailor their efforts and prioritize high-value prospects. Plus, it may

even help spotlight new opportunities for the business to pursue.

Sales analytics allow us to better gauge team performance and uncover areas for improvement, too. Understanding those strengths and weaknesses leads to better training, more attainable goals, and a cohesive team.

It allow the sales team to know the following metrics:

- **Sales growth**

  Sales growth shows how much your revenue increases (or decreases) over a specific period of time. This metric provides a bird's-eye view of sales and how your team is performing.

- **Sales target**

  Sales target evaluates current sales and compares them to your bigger, long-term goals.

- **Sales per rep**

  Sales per rep measures the individual performance of agents.

- **Sales by region**

  Sales by region dives into the volume of sales in key geographical areas for the business.

- **Sales per product**

  Sales per product, also called product performance, shows the profitability of each product that sells.

- **Sales through rate**

  The sell-through rate assesses how quickly a business can sell its inventory.

LITERATURE SURVEY

2.1 EXISTING PROBLEM

The problem that every companies are facing is to analyze the dataset of sales performance of any company , if a single app can analyze and plot the graph of the dataset and if it provides a reliable predicted future sales means it can be a very helpful one to every companies

2.2 REFERENCES

- WALMART'S SALES DATA ANALYSIS- A BIG DATA ANALYTICS PERSPECTIVE – 2019

  (Manpreet Singh, Bhawick Ghutla, Reuben Lilo Jnr, Aesaan F S Mohammed, Mahmood A Rashid)

- Analytics-driven solutions for customer targeting and salesforce allocation - 2007 (R. Lawrence C. Perl ich)

- DATA-DRIVEN SALES LEADS PREDICTION FOR EVERYTHING AS A SERVICE IN THE CLOUD - 2022

(Chul Sung, Bo Zhang, Chunhui Y. Higgins Yoonsuck Choe)

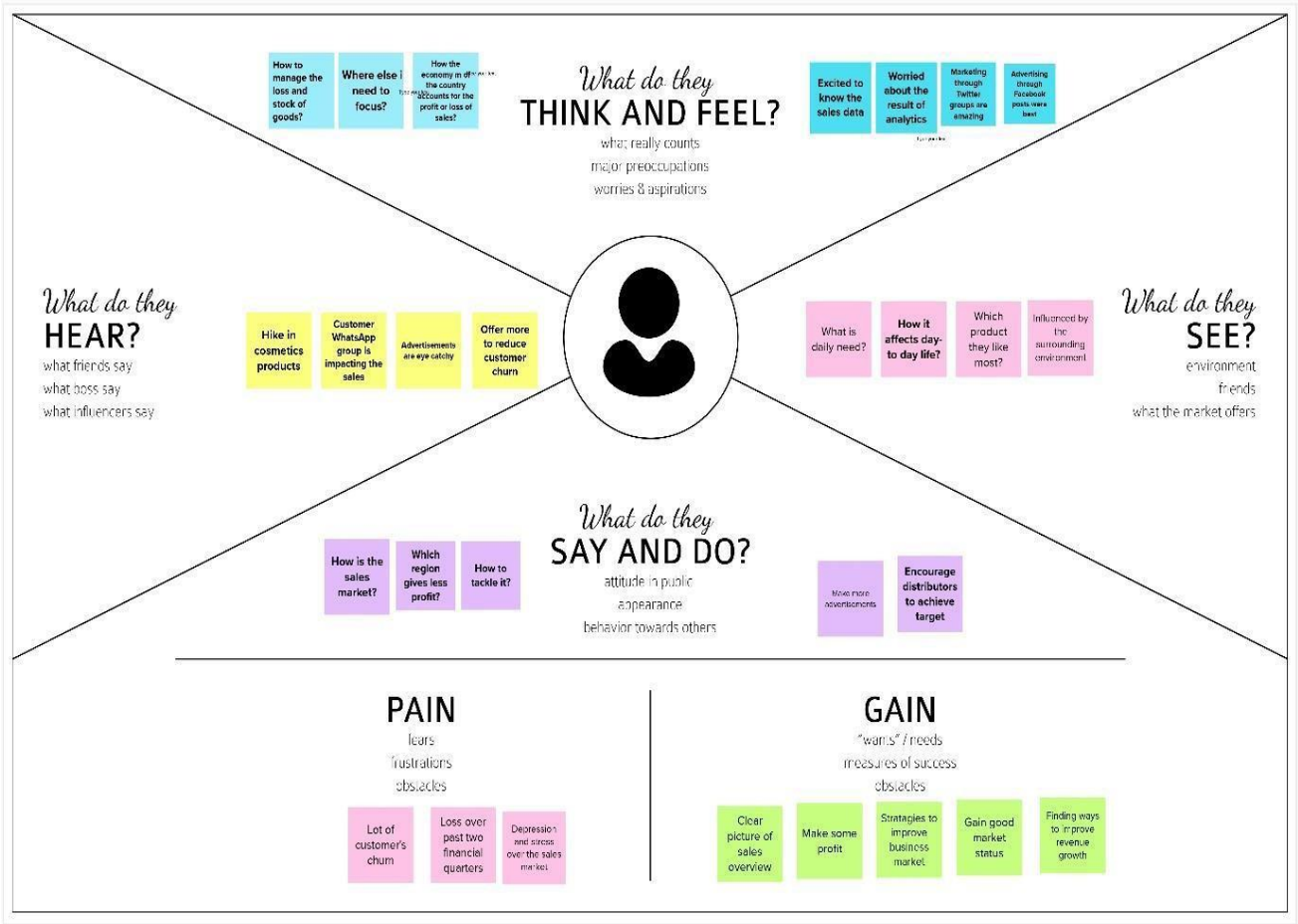
- Analysis and Optimization of Online Sales of Products - 2017 (Z. Pirani, A. Marewar, Z. Bhavnagarwala and M. Kamble)

2.3 PROBLEM STATEMENT

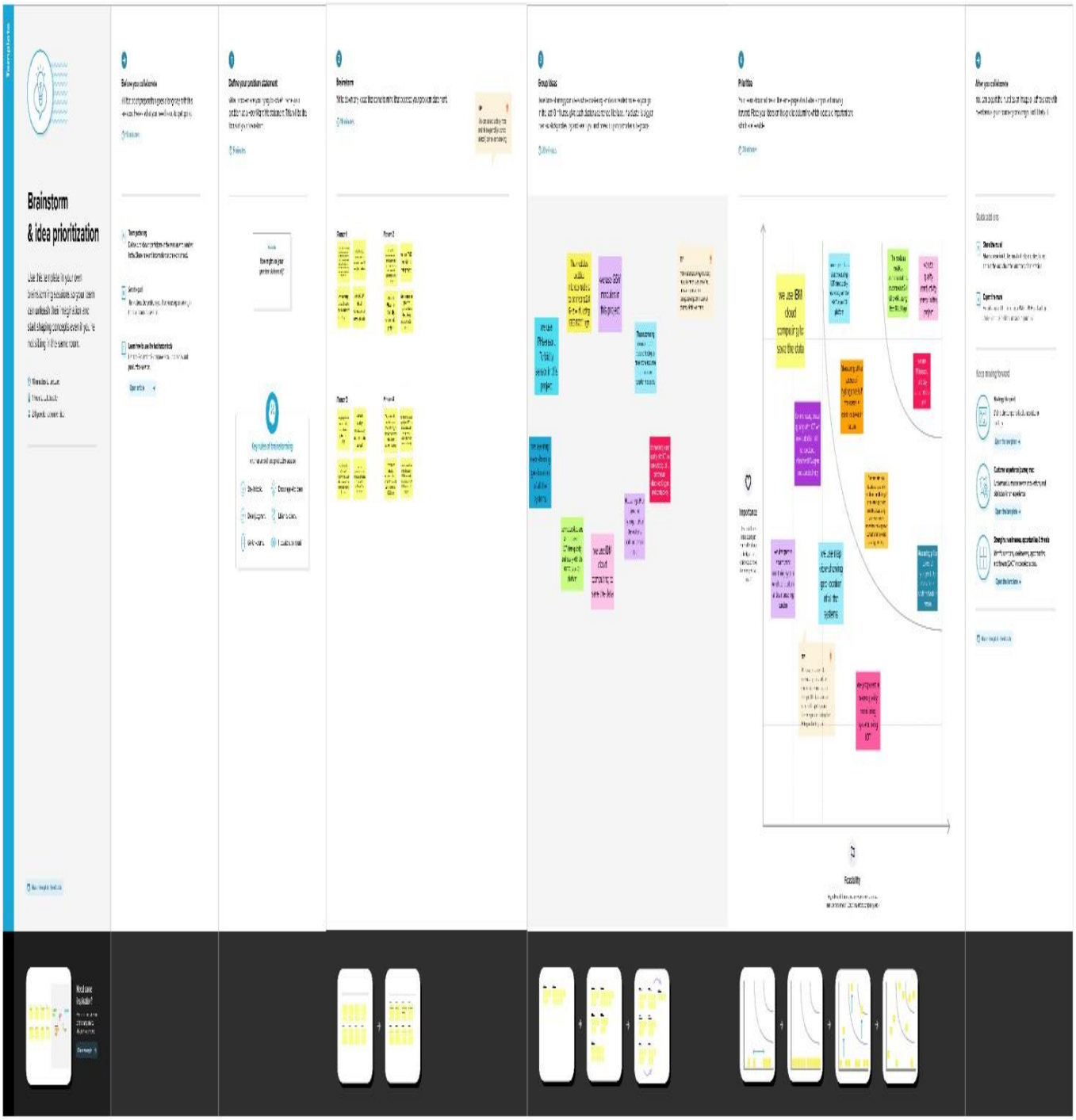
PROBLEM STATEMENT	I'M (CUSTOMER)	I'M TRYING TO	BUT	BECAUSE	WHICH MAKES ME FEEL
PS-1	SHOP OWNER	KNOW SALES RATE IS STABLE	I'M JUST HAVING THE DATASET	I WANT TO SURVIVE IN THE MARKET	SAD
PS-2	SALES TEAM HEAD	IMPROVE THE SALES RATE	I'M JUST HAVING THE DATASET	THE COMPANY CAN'T OFFER HIM PROMOTION	FRUSTATED

3 IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Shopping online is currently the need of the hour. Because of this COVID, it's not easy to walk in a store randomly and buy anything you want. So, try to understand a few things like, Customer Analysis and Product Analysis of this Global Super Store.
2.	Idea / Solution description	Analyse sales by demographic Analysis of customers Ex: city, age, gender. The goal of this process is to give more information about our data so that the marketing team prepares to intensify the efficiency based on the data.
3.	Novelty / Uniqueness	• Improve Value Propositions and Price Points • Narrow and Refine Product Offerings • Accurate Sales Forecasting
4.	Social Impact / Customer Satisfaction	By analysing the maximum products sales, we can keep in stock in the markets so that the people will not get affected in buying the needed products.
5.	Business Model (Revenue Model)	By analysing the sales data, the company starts generating the specific products more. By this, the company profit margin of these products gets more.
6.	Scalability of the Solution	Analysing the data continuously to improve the efficiency of the products.

3.4 PROBLEM SOLUTION

1.CUSTOMER SEGMENT

People and corporations who are interested in knowing the details and a complete global sales analysis of a product and related products for effective customer making organizations which wants to know their products sales details.

6.CUSTOMER LIMITATIONS

Lack of understanding over the business and the customer engagement over it

5.AVAILABLE SOLUTIONS

Sales metrics, revenues, gross number of sales. Simply measuring revenue or the gross number of sales isn't enough. The right metrics will depend on your company, but are valuable to learn more about your company, customers, and sales process.

2.PROBLEMS/PAINS

Sales Analysis provides insights into the past, present, and future performance of a business and can be used to help you forecast trends, identify opportunities for growth, and develop a strategic action plan for your company.

9.PROBLEM ROOT/CAUSE

Being lethargic that the business is doing fine, absence of customer involvement monitoring, presence and progress of competitors in global market.

7.BEHAVIOUR

Actions against losing customers, changes in budget, advertising and collaborations for betterment.

3.TRIGGERS TO ACT

Better performance of their business competitors, fall down of their performance.

4.EMOTIONS

Satisfaction and may lead to new achievement and betterment of self and business growth.

10.YOUR SOLUTION

- Creating an interactive dashboard.
- Providing specific details about sales
- Responsive design for every screen size
- Manual insight for each interaction.
- One time payment

8.CHANNEL OF BEHAVIOUR

8.1 ONLINE

Using third party services with automated insights and subscription based service to analyse data.

8.2 OFFLINE

Using office software to analyse complex data in un-intuitive way.

4 REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

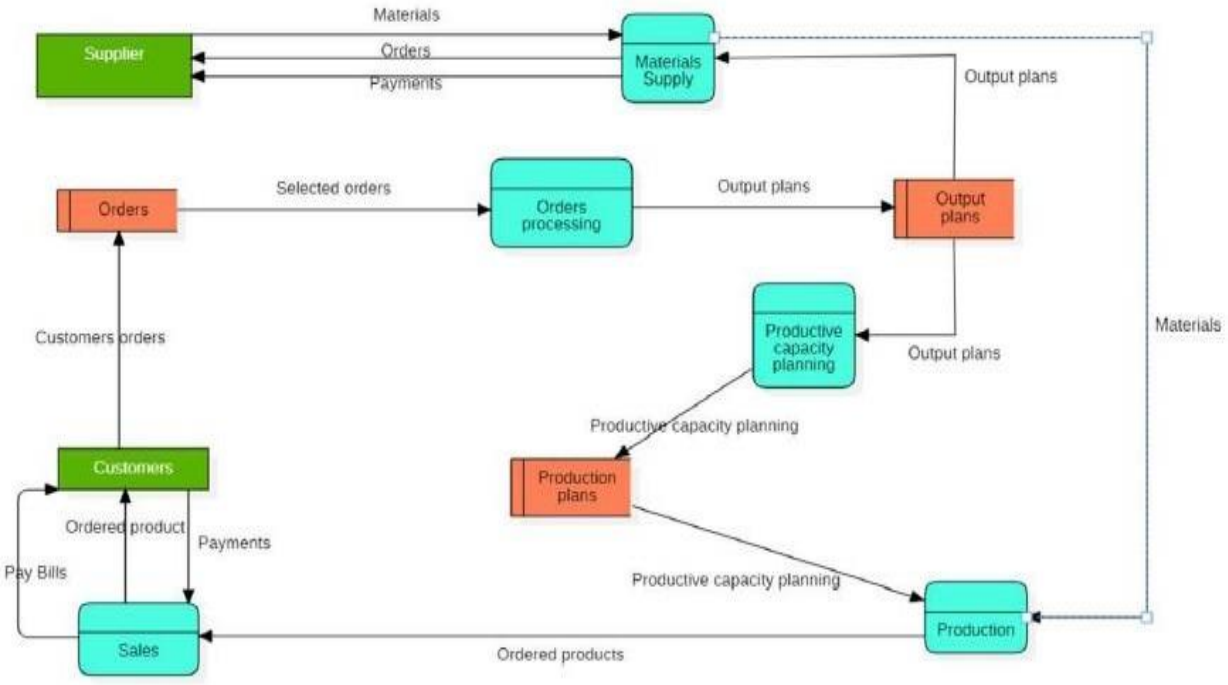
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User dashboard creation	Created by ibm cognos analytics
FR-4	Predictive analytics	Done by using Machine Learning in python programming language
FR-5	Creating visualization	Created using IBM cognos analytics

4.2 NON FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Used to understand data and analyse the data by visuzlization
NFR-2	**Security**	Security depends on the data and the analyst
NFR-3	**Reliability**	The data can be updated and the task can be executed.
NFR-4	**Performance**	Performance is high were large amont of data can be used effectively
NFR-5	**Availability**	Available at a time and anywhere when upload to cloud.
NFR-6	**Scalability**	Scalability depends on the size of the data in a task.

5 PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Technical Architecture:

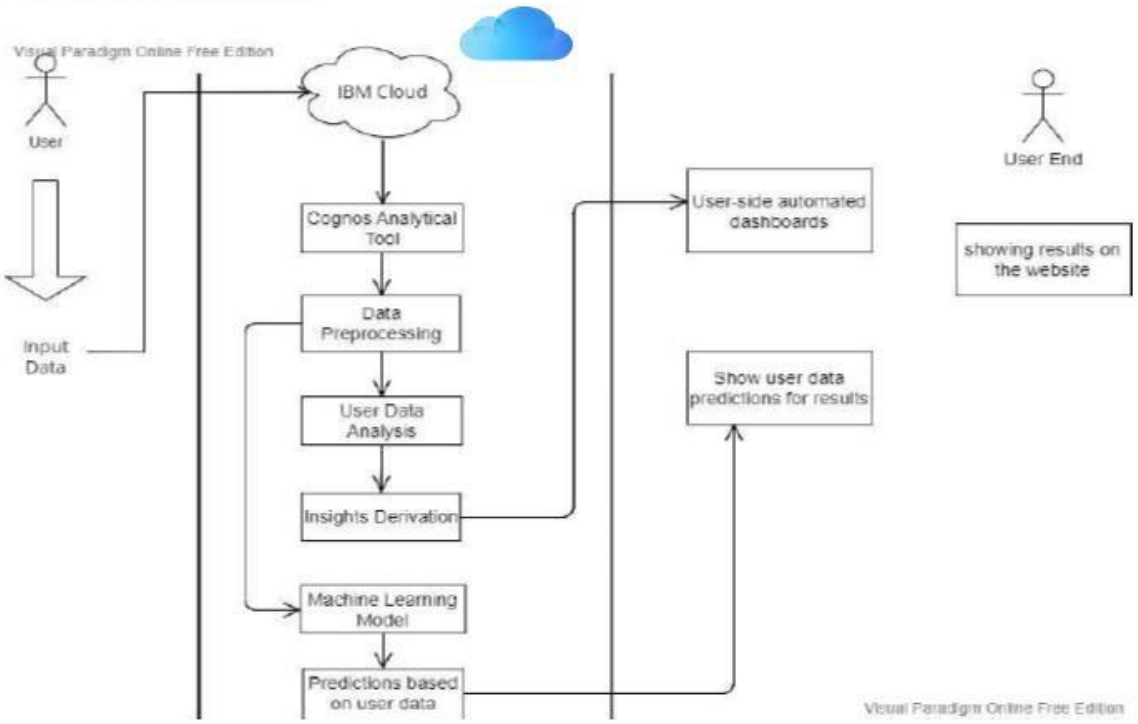


Table-1: Components & Technologies:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Google Collaboratory, Jupyter notebook	Google
2.	Security Implementations	To protect data from the unauthorized access	256-bit AES algorithm
3.	Scalable Architecture	Supports various data sizes	IBM Cloud

S.No.	Component	Description	Technology
1.	User Interface	User uploads the csv or excel format files into the web pages	HTML, CSS, JavaScript
2.	Application Logic-1	The user data will pass into the IBM cloud for storing and acts as a data source	IBM cloud
3.	Application Logic-2	In cloud, data will be fetched by the Cognos analytical tool for data analysis	IBM Cognos analytical tool
4.	Application Logic-3	The pre-trained Dashboards will be present to perform analysis on the incoming data	IBM Cognos analytical tool
5.	Database	Data will be retrieved from cloud	MySQL
6.	Cloud Database	Database Service on cloud	IBM DB2, IBM Cloud
7.	File Storage	Customer sales data is uploaded in cloud through interface	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	To perform data analysis on the user data	IBM Cognos Tool
9.	External API-2	To build the machine learning model for classification	Jupiter Notebook
10.	Machine Learning Model	To do the predictive analysis on the input data	Predictive analysis model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Using the flask Cloud Server Configuration: IBM cloud	Local, Cloud Foundry

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
4.	Availability	Multi page layout providing various visualizations of data and provide full support irrespective of platform and device specifications	Cognos Business Intelligence Server
5.	Performance	Withstand huge data and process them without crashing	IBM Cognos

5.3 USER STORIES

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / Dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user, I can create the visualization by using the dashboard in the application		High	Sprint-3

Customer (Web user)	Login	USN-1	As a user,I can register for the application by entering my email ,password and confirming my password	I can access my account and dashboard	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer Care Executive	Chat box	USN-1	It can be used by easily access and responsible.	I can access by easily through application	High	Sprint-2
Administrator	Calling	USN-2	It can be used by easily access and responsible.	I can access by easily through application	High	Sprint-2
	Mail	USN-3	It can be used by easily access and responsible	I can access by easily through application	High	Sprint-1

6 PROJECT PLANNING

6.1 SPRINT PLANNING AND SCHEDULING

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	user can register for the application by entering my email and password	1	High	SHYAM J
Sprint-1	Registration	USN-2	User will receive email if the registration is successful. That the registration has conformed	1	High	SHYAM J
Sprint-2	Registration	USN-3	As a user, I can register by any browser.	2	Low	SARAVANAN B
Sprint-1	Data extract	USN-4	As a user, I can extract data	1	Medium	RAGUL RAM PRASHANTH A
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	2	High	NIRMALRAJ K
Sprint-2	Dashboard	USN-6	I can access the dashboard of mine.	1	Medium	SATHISH KUMAR S
Sprint-1	Activity	USN-7	I can register for the application through any web browser.	1	low	NIRMALRAJ K
Sprint-1	Access resources	USN-8	I can use my credentials For accessing my resources.	1	high	SARAVANAN B
Sprint-2	Set events	USN-9	As, a user I can schedule events and set events.	1	high	SHYAM J
Sprint-3	Tools	USN-10	I can perform analysis by tools (cognos and with ML)	1	high	SRIMANJUNATH D

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7 CODING AND SOLUTIONING

7.1 FEATURE 1

#DATA PREPROCESSING

```
import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
```

#Reading Dataset File

```
store_sales=pd.read_csv("train.csv")
```

#Dropping Unneccessary Coloumn

```
store_sales=store_sales.drop(['store','item'],axis=1

#converting date from object datatype to dateTime datatype

store_sales['date']=pd.to_datetime(store_sales['date'])


#Converting date to a month period and sum the number of items of each month


store_sales['date']=store_sales['date'].dt.to_period('M')

monthly_sales=store_sales.groupby('date').sum().reset_index()


#Converting the resulting data coloumn into timeStamp datatype


monthly_sales['date']=monthly_sales['date'].dt.to_timestamp()
```

#VISUALIZATION OF MONTHLY SALES AND MONTHLY SALES DIFFERENCE BY YEAR

```
import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt



#Visualisation



plt.figure(figsize=(15,5))

plt.plot(monthly_sales['date'],monthly_sales['sales'])

plt.xlabel('date')

plt.xlabel('sales')
```

```
plt.title('Monthly Customer Sales')

plt.show()



monthly_sales['sales_diff']=monthly_sales['sales'].diff()

monthly_sales=monthly_sales.dropna()

plt.figure(figsize=(15,5))

plt.plot(monthly_sales['date'],monthly_sales['sales'])

plt.xlabel('date')

plt.xlabel('sales')

plt.title('Monthly Customer Sales Difference')

plt.show()

#PREPARING THE SUPERVISED DATA TO GIVE IT TO THE

MACHINE LEARNING MODEL
```

```
import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from xgboost import XGBRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.linear_model import LinearRegression

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import

mean_absolute_error,mean_squared_error,r2_score



supervised_data=monthly_sales.drop(['date','sales'],axis=1)



#preparing The Supervised Dataset
```

```
for i in range(1,13):
  col_name='month'+str(i)
  supervised_data[col_name]=supervised_data['sales_diff'].shift(i)
supervised_data=supervised_data.dropna().reset_index(drop=True)
```

7.2 FEATURE 2

```
#Testing and Training Data

train_data=supervised_data[:-12]
test_data=supervised_data[-12:]
```

```
scaler=MinMaxScaler(feature_range=(-1,1))

scaler.fit(train_data)

train_data=scaler.transform(train_data)

test_data=scaler.transform(test_data)




x_train,y_train=train_data[:,1:],train_data[:,0:1]

x_test,y_test=test_data[:,1:],test_data[:,0:1]

y_train=y_train.ravel()

y_test=y_test.ravel()



#Make prediction dataframe to merge the predicted sales price of all

trained algorithms
```

```
sales_dates=monthly_sales['date'][-12:].reset_index(drop=True)

predict_df=pd.DataFrame(sales_dates)



act_sales=monthly_sales['sales'][-13:].to_list()

#TRAINING THE LINEAR REGRESSION MODEL AND

PLOTTING THE FORECAST GRAPH



import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from xgboost import XGBRegressor

from sklearn.ensemble import RandomForestRegressor
```

```
from sklearn.linear_model import LinearRegression

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import

mean_absolute_error,mean_squared_error,r2_score

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense ,LSTM

from tensorflow.keras.callbacks import EarlyStopping ,

ModelCheckpoint



#To Creat Linear Regression Model and Predict The Output



lr_model=LinearRegression()

lr_model.fit(x_train,y_train)

lr_pre=lr_model.predict(x_test)
```

```
lr_pre=lr_pre.reshape(-1,1)

lr_pre_test_set=np.concatenate([lr_pre,x_test],axis=1)

lr_pre_test_set=scaler.inverse_transform(lr_pre_test_set)



result_list=[]

for index in range(0,len(lr_pre_test_set)):

result_list.append(lr_pre_test_set[index][0]+act_sales[index])

lr_pre_series=pd.Series(result_list,name='Linear Prediction')

predict_df=predict_df.merge(lr_pre_series,left_index=True,right_in

dex=True)
```

```
lr_mse=np.sqrt(mean_squared_error(predict_df['Linear
Prediction'],monthly_sales['sales'][-12:]))

lr_mae=mean_absolute_error(predict_df['Linear
Prediction'],monthly_sales['sales'][-12:])

lr_r2=r2_score=(predict_df['Linear
Prediction'],monthly_sales['sales'][-12:])



#Visualization of the prediction against the actual sales



plt.figure(figsize=(15,5))



#Actual Sales



plt.plot(monthly_sales['date'],monthly_sales['sales'])
```

#Predicted Sales

```
plt.plot(predict_df['date'],predict_df['Linear Prediction'])

plt.title('Customer sales Forecast Using LR Model')

plt.xlabel('date')

plt.ylabel('sales')

plt.legend(['Actual Sales','predicted Sales'])

plt.show()
```

8 TESTING

8.1 TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	2	0	0	2
Client Application	2	0	0	2
Security	2	0	0	2
Outsource Shipping	2	0	0	2

Exception Reporting	2	0	0	2
Final Report Output	2	0	0	2
Version Control	2	0	0	2

8.2 USER ACCEPTANCE TESTING

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
Getting File Name As a Input From The User	Functional	Home Page	Verify user is able to give the test dataset name		Entering the file name	provided in final deliverables	File should be read without error	Working as expected	Pass
Getting The File Path As a Input From The User	Functional	Home Page	Verify the User is able to give the file path as a input without any error		Entering the file path	Provided in Final Deliverables	File should be read withot error	Working as expected	pass
Data preprocessing	Functional	Home page	Verify that the program is able to clear the unwanted data and add the additional data to go through with the further process and to increase the accuracy of the model		1.Enter the file path or file name	Provided on final deliverables	Data preprocessing should be done without error	Working as expected	pass
Data Visualization	Functional	Home Page	Verify the user is able to visualize the input sales details by month		1.Enter the file path or file name	Provided on final deliverables	Input data visualization should be done successfully	Working as expected	pass
Data Preparation To Fit For The Model	Functional	Home Page	Verify the program is able to transform the given dataset to fit for the Linear Regression		1.Enter the file path or file name	Provided on final deliverables	Data tranformation for feeding to the model should be done without error	Working as expected	pass
To Find The Relation	Functional	Home page	Verify the user is user is able to find out the relation		1.Enter the file path or file name	Provided on final deliverables	Training of machine learning model(Linear regression model)	Working as expected	pass
Visualization Of The Pedicted Sales By Year	Functional	Home Page	Verify the user is able to see the visualiztion of the predicted sales by year without any errror		1.Enter the file path or file name	Provided on final deliverables	Plotting of the predicted data should be done successfully	Working as expected	pass

9 RESULT

9.1 PERFORMANCE METRICS

PARAMETER	PERFORMANCE	DESCRIPTION
ADMIN TESTING	95%-100%	THE TESTING DONE BEFORE IT IS DEPLOYED AS AN APP
CUSTOMER SATISFACTION	75-85%	THE CUSTOMER NEED TO BE SATISFIED WITH THE MOBILE APPLICATION
USER INTERFACE	65-85%	THE APP CAN USED BY ANYONE.(EASE OF ACCESS)

SEVER RESPONSE	50-75%	url - response
DATA VALIDATION WITH NO. OF TEST CASE	60-80% (15-30 TESTCASE)	VALID DATA FROM THE APP
ERROR	3-5%	REAL-TIME DELAY MAY OCCUR

10 ADVANTAGES AND DISADVANTAGES

- **OUR PROJECT CAN PREDICT THE FUTURE SALES RATE**
- **WE CAN ALSO VISUALIZE IT WITH THE GRAPH WHICH CAN BE VERY EASY TO UNDERSTAND**
- **THE MACHINE LEARNING MODEL WHICH WE HAVE PROVIDED CAN MAKE SOME MISMATCH WITH PREDICTED DATA AND THE ORIGINAL DATA**
- **IF K VALUE IS LESSER OR MORE HIGHER IT IS NOT RECOMMENDED TO USE THIS PROGRAM FOR THAT DATASET**

11 CONCLUSION

OUR PROJECT IS DESIGNED TO USE IN THE DATASET FROM THE USER MAYBE A SHOP OR COMPANY MANAGER. BY GETTING A DATA IT WILL PROCESS IT BY USING THE MACHINE LEARNING MODEL AND PROVIDE US THE PREDICTED SALES RATE WITH THE HELP OF THE PAST

SALES RATE WE CAN ALSO MODIFY THE CODE ACCODING TO THE USER NEEDS THAT THE USER THINK THAT THE PARAMETERS THAT THEY ARE PROVIDING I.E,, INPUTS MAY BE MORE DETAILED WITH SOME MORE DATA

12 FUTURE SCOPE

THE OLD METHOD THAT THE MANAGEMENT FOLLOWING TO MANIPULATE WITH THE DATA ARE NO SO EFFICIENT AND IT TAKE MORE TIME TO EXTRACT SOME IDEA ABOUT IT .THE PROGRAM WE HAVE CREATED WILL HELP THEM TO PROCESS THE DATA WITH EASE

13 APPENDIX

13.1 SOURCE CODE

#DATA PREPROCESSING

```
import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt



#Reading Dataset File



store_sales=pd.read_csv("train.csv")

#Dropping Unneccessary Coloumn

store_sales=store_sales.drop(['store','item'],axis=1

#converting date from object datatype to dateTime datatype

store_sales['date']=pd.to_datetime(store_sales['date'])
```

#Converting date to a month period and sum the number of items of each month

```
store_sales['date']=store_sales['date'].dt.to_period('M')

monthly_sales=store_sales.groupby('date').sum().reset_index()
```

#Converting the resulting data coloumn into timeStamp datatype

```
monthly_sales['date']=monthly_sales['date'].dt.to_timestamp()
```

#VISUALIZATION OF MONTHLY SALES AND MONTHLY SALES DIFFERENCE BY YEAR

#Visualisation

```
plt.figure(figsize=(15,5))

plt.plot(monthly_sales['date'],monthly_sales['sales'])

plt.xlabel('date')

plt.xlabel('sales')

plt.title('Monthly Customer Sales')

plt.show()



monthly_sales['sales_diff']=monthly_sales['sales'].diff()

monthly_sales=monthly_sales.dropna()

plt.figure(figsize=(15,5))

plt.plot(monthly_sales['date'],monthly_sales['sales'])

plt.xlabel('date')
```

```
plt.xlabel('sales')

plt.title('Monthly Customer Sales Difference')

plt.show()

#PREPARING THE SUPERVISED DATA TO GIVE IT TO THE
MACHINE LEARNING MODEL


from xgboost import XGBRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.linear_model import LinearRegression

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import
mean_absolute_error,mean_squared_error,r2_score


supervised_data=monthly_sales.drop(['date','sales'],axis=1)
```

```
#preparing The Supervised Dataset


for i in range(1,13):
  col_name='month'+str(i)
  supervised_data[col_name]=supervised_data['sales_diff'].shift(i)
supervised_data=supervised_data.dropna().reset_index(drop=True)
#Testing and Training Data


train_data=supervised_data[:-12]
test_data=supervised_data[-12:]
```

```
scaler=MinMaxScaler(feature_range=(-1,1))

scaler.fit(train_data)

train_data=scaler.transform(train_data)

test_data=scaler.transform(test_data)




x_train,y_train=train_data[:,1:],train_data[:,0:1]

x_test,y_test=test_data[:,1:],test_data[:,0:1]

y_train=y_train.ravel()

y_test=y_test.ravel()



#Make prediction dataframe to merge the predicted sales price of all

trained algorithms
```

```
sales_dates=monthly_sales['date'][-12:].reset_index(drop=True)

predict_df=pd.DataFrame(sales_dates)



act_sales=monthly_sales['sales'][-13:].to_list()

#TRAINING THE LINEAR REGRESSION MODEL AND

PLOTTING THE FORECAST GRAPH



import pandas as pd

from xgboost import XGBRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.linear_model import LinearRegression

from sklearn.preprocessing import MinMaxScaler
```

```
from sklearn.metrics import
mean_absolute_error,mean_squared_error,r2_score
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense ,LSTM
from tensorflow.keras.callbacks import EarlyStopping ,
ModelCheckpoint


#To Create Linear Regression Model and Predict The Output


lr_model=LinearRegression()
lr_model.fit(x_train,y_train)
lr_pre=lr_model.predict(x_test)


lr_pre=lr_pre.reshape(-1,1)
```

```
lr_pre_test_set=np.concatenate([lr_pre,x_test],axis=1)

lr_pre_test_set=scaler.inverse_transform(lr_pre_test_set)



result_list=[]

for index in range(0,len(lr_pre_test_set)):

result_list.append(lr_pre_test_set[index][0]+act_sales[index])

lr_pre_series=pd.Series(result_list,name='Linear Prediction')

predict_df=predict_df.merge(lr_pre_series,left_index=True,right_in
dex=True)



lr_mse=np.sqrt(mean_squared_error(predict_df['Linear
Prediction'],monthly_sales['sales'][-12:]))
```

```
lr_mae=mean_absolute_error(predict_df['Linear
Prediction'],monthly_sales['sales'][-12:])

lr_r2=r2_score=(predict_df['Linear
Prediction'],monthly_sales['sales'][-12:])



#Visualization of the prediction against the actual sales



plt.figure(figsize=(15,5))



#Actual Sales



plt.plot(monthly_sales['date'],monthly_sales['sales'])



#Predicted Sales
```

plt.plot(predict_df['date'],predict_df['Linear Prediction'])

plt.title('Customer sales Forecast Using LR Model')

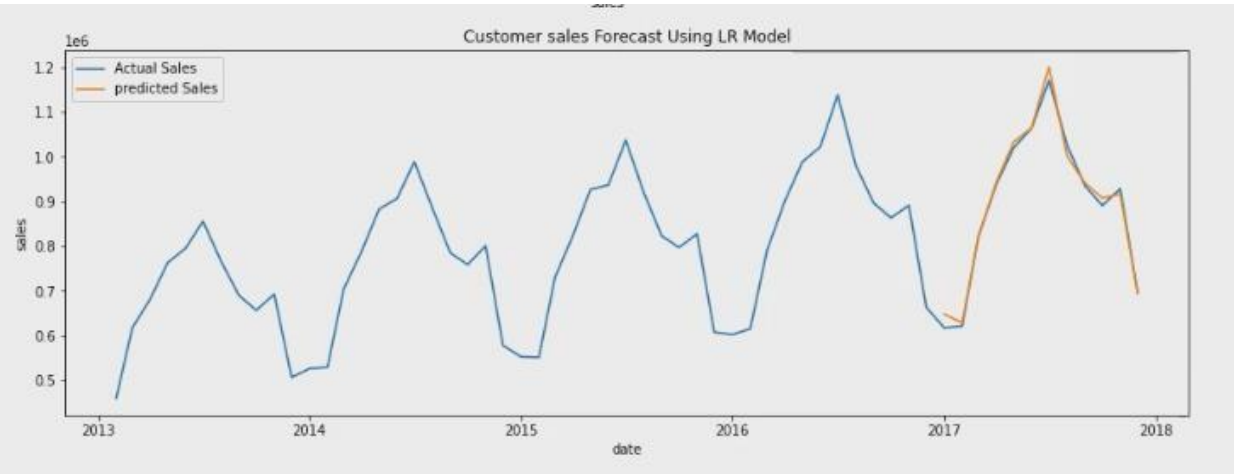
plt.xlabel('date')

plt.ylabel('sales')

plt.legend(['Actual Sales','predicted Sales'])

plt.show()

13.2 OUTPUT



GitHub LINK:

https://github.com/IBM-EPBL/SI-GuidedProject-53111-1665749846

DEMO LINK:

https://youtu.be/90uLywEkJy8