National University
of computer and emerging sciences

(KARACHI CAMPUS)
Department of Computer Science
Fall 2020

[Project Final Report]

Project Title:

ATM Cash Predictor

[To predict ATM withdrawal patterns in order to calculate the optimal sum currency to be made available thereof]

By:

Muhammad Owais Mushtaq (18K-1177)

Muhammad Usman Umar (18k-1069)

Shaharyar Amjad (18k-1371)

INDEX

ACKNOWLEDGEMENT

It is indeed with a great pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We are highly indebted to our Director **Dr. Jawwad Shamsi,** FAST-National University of Computer and Emerging Sciences, for the facilities provided to accomplish this main project.

We would like to thank our **Dr. Zulfiqar Memon**, Head of the Department of Computer Science, FAST-National University of Computer and Emerging Sciences, for this constructive criticism throughout our project.

We feel elated in manifesting our sense of gratitude tour internal project guide **Dr. Fahad Shirwani, Assistant Profesor, Department of Computer Science**, FAST-National University of Computer and Emerging Sciences and respected **Ma'am Nimra Iqbal, Lecturer, Department of Computer Science**, FAST-National University of Computer and Emerging Sciences. She has been a constant source of inspiration for us and we are very deeply thankful to him for his support and valuable advice.

We extremely grateful to our Departmental staff members, Lab technicians and Non-teaching staff members for their extreme help throughout our project.

Project Associates:

Muhammad Owais Mushtaq (18K-1177)

Muhammad Usman Umar (18K-1069)

Shaharyar Amjad (18K-1371)

DECLARATION

We hereby declare that project titled **“ATM Cash Predictor”** is an original record done by us at **FAST-NUCES,** towards the partial fulfillment of requirement for the award of degree of Bachelor of Computer Science during the period of 2018-2022 in **FAST-NUCES**, and also we state that this project has not been submitted anywhere in the partial fulfillment for any degree of this or any other University.

Muhammad Owais Mushtaq (18K-1177)
Muhammad Usman Umar (18K-1069)
Shaharyar Amjad (18K-1371)

ABSTRACT

Cash demand in ATMs requires accurate prediction because if the forecast is wrong, it induces a considerable amount of costs. Although the banks pays a significant amount of fixed fees for the refiling and addition cost for transportation but in case of high forecast and high unused cash stored in the ATMs might be save refiling or transport expenses but actually it causes the loss of bank because Bank tries to invest each penny so it can gain profit or commission. The purpose of the project entitled as "**ATM Cash Predictor**" is to predict the correct amount of cash that will filled on the next day in ATM so that Banks can earn more by not storing extra money on ATMs. To accomplish our goal we are using some very famous Machine Learning Techniques called as Linear Regression and it's an advance version XGB Boost to train and test our models.

INTRODUCTION

4.1. Introduction:

ATM should not be filled with large amount of cash which may bring low transport/logistic cost but high freezing and high insurance costs. The purpose of the project entitled as "**ATM Cash Predictor**" is to predict the correct amount of cash that will filled on the next day in ATM so that Banks can earn more by not storing extra money on ATMs. To accomplish our goal we are using some very famous Machine Learning Techniques called as Linear Regression and it's an advance version XGB Boost to train and test our models.

4.2. Problem Introduction:

Cash demand in ATMs requires accurate prediction because if the forecast is wrong, it induces a considerable amount of costs. Although the banks pays a significant amount of fixed fees for the refiling and addition cost for transportation but in case of high forecast and high unused cash stored in the ATMs might be save refiling or transport expenses but actually it causes the loss of bank because Bank tries to invest each penny so it can gain profit or commission.

4.3. Scope of the Project:

Prediction could be failed in some special event such as:

- Before the Eid people withdraws lot of money in unexpected order form ATMs because people do shopping etc.
- But some time like this year due to lockdown not many people withdrawal money form ATM's because most of the markets were close.
- The location of ATM machine also matters, ATM's in big city needs more money than ATM's in small towns. Hence the change of location of any ATM may be cause the wrong prediction.

REQUIREMENT SPECIFICATION

5.1. INTRODUCTION:

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as (computer) system requirements and are often used as a guideline rather than an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

5.2. HARDWARE REQUIREMENTS:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

NORMAL HARDWARE REQUIREMENTS FOR PRESENT PROJECT:

PROCESSOR	:	Intel dual Corei3
RAM	:	1 GB
HARD DISK	:	80 GB

5.3. SOFTWARE REQUIREMENTS:

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

SOFTWARE REQUIREMENTS FOR PRESENT PROJECT:

OPERATING SYSTEM : Windows 7/8/10, Linux
Compiler: Anaconda's Jupyter Notebook
Language: Python

5.4. Functional Requirements:

Statements of services the system should provide and how the system should react to particular inputs and in particular situations. May state what the system should not do.

Functional Requirements

Our System has following functional requirements:

- Predict Cash for next day of ATM
- Accurate Prediction
- Train model on 80 % of previous data record of ATM
- Test model on the rest of 20% of same dataset
- Visualize dataset

5.5 Non-Functional Requirements:

Non-Functional Requirements specifies the quality attribute of a software system. They judge the software system based on Usability, Security, Localization, Responsiveness, Portability, Compatibility and other non-functional standards that are critical to the success of the software system.

Non-Functional Requirements

Our System has following non-functional requirements:

- User friendly GUI (Not matters in our case)
- Performance (e.g. compiler loading time, Dataset uploading time etc.)

ANALYSIS

6.1. EXISTING SYSTEM:

There are many system has been lunched all over the world that provides accurate predictions of cash for next day filling of ATM, each bank has its own unique software. Many companies all around the world develops the unique ATM prediction soft wares on order.

6.2. PROPOSED SYSTEM:

Our proposed system **ATM Cash Predictor** work on Linear Regression and it's an advance version XGB Boost to train and test our models. It divides the dataset into two parts: one of 80% for training data and the other one of 20% for testing data. Usually it uses Linear Regression but if the accuracy is not coming accurate from this method then it switches to it's an advance version which is XG Boost which can give the 80 to 85 percent accuracy of that data for which linear regression proposed 50 %.

6.3. FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

6.3.1. Economic Feasibility

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products have to be purchased.

6.3.2. Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

6.3.3. Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

6.4. SOFTWARE SPECIFICATION

Jupyter Notebook:

Jupyter Notebook is a sub application or part of Anaconda. It’s a web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

SYSTEM IMPLEMENTATION

7.1. SYSTEM DESIGN:

7.1.1. INTRODUCTION TO UML:

UML Design

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

- Visualizing
- Specifying
- Constructing
- Documenting

7.2. UML Approach

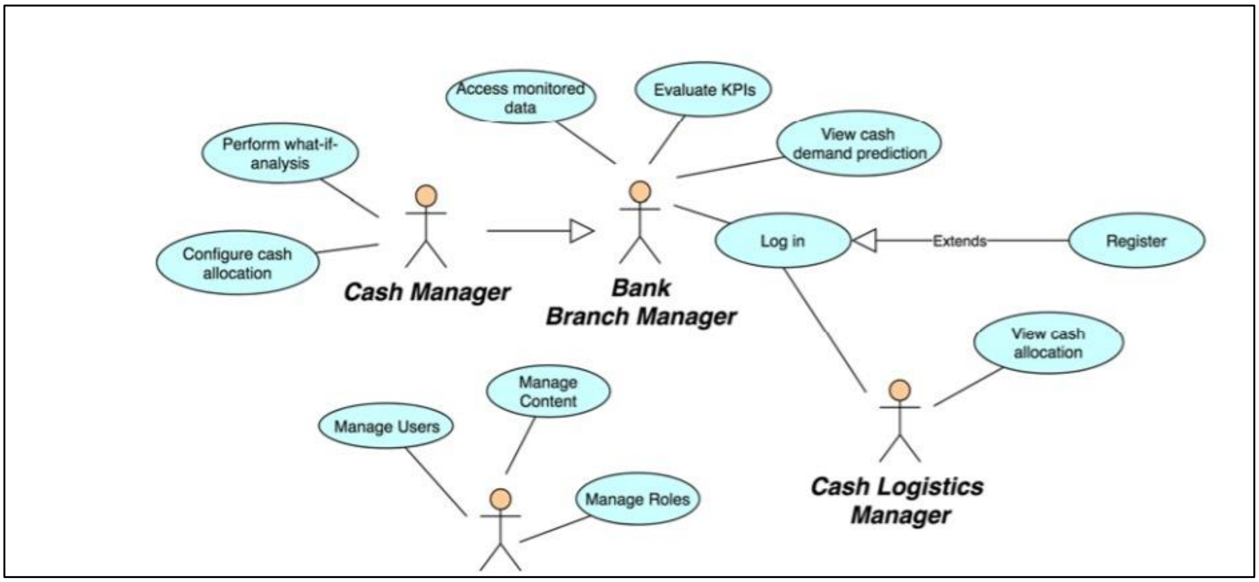
UML Diagram

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs. You draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams, or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

i. Use case diagram
ii. Activity diagram
iii. Sequence diagram
iv. Collaboration diagram
v. State chart diagram
vi. Component diagram
vii. Class diagram
viii. Object diagram
ix. Deployment diagram

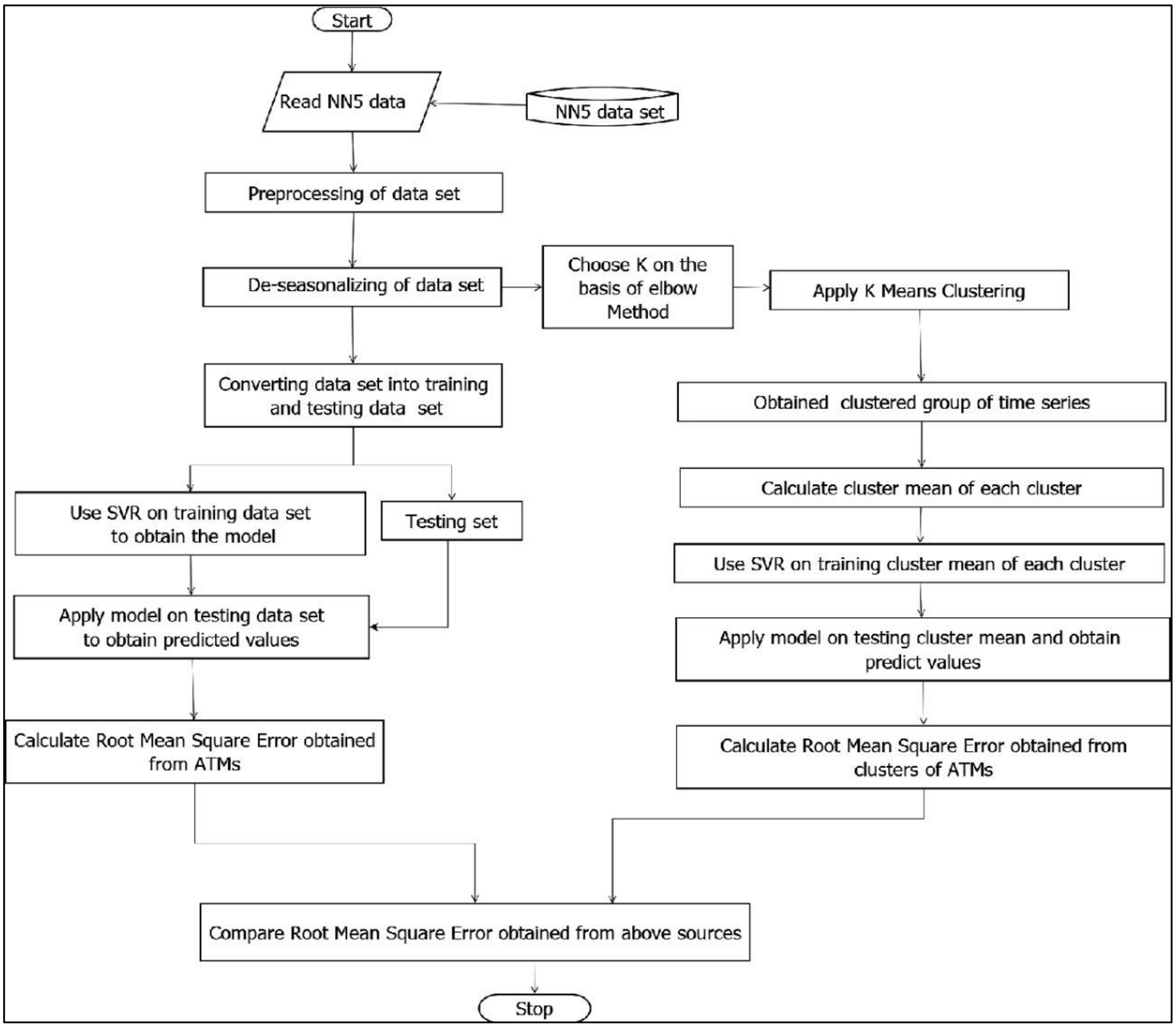
7.2.1. USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.



7.2.2. ACTIVITY DIAGRAM:

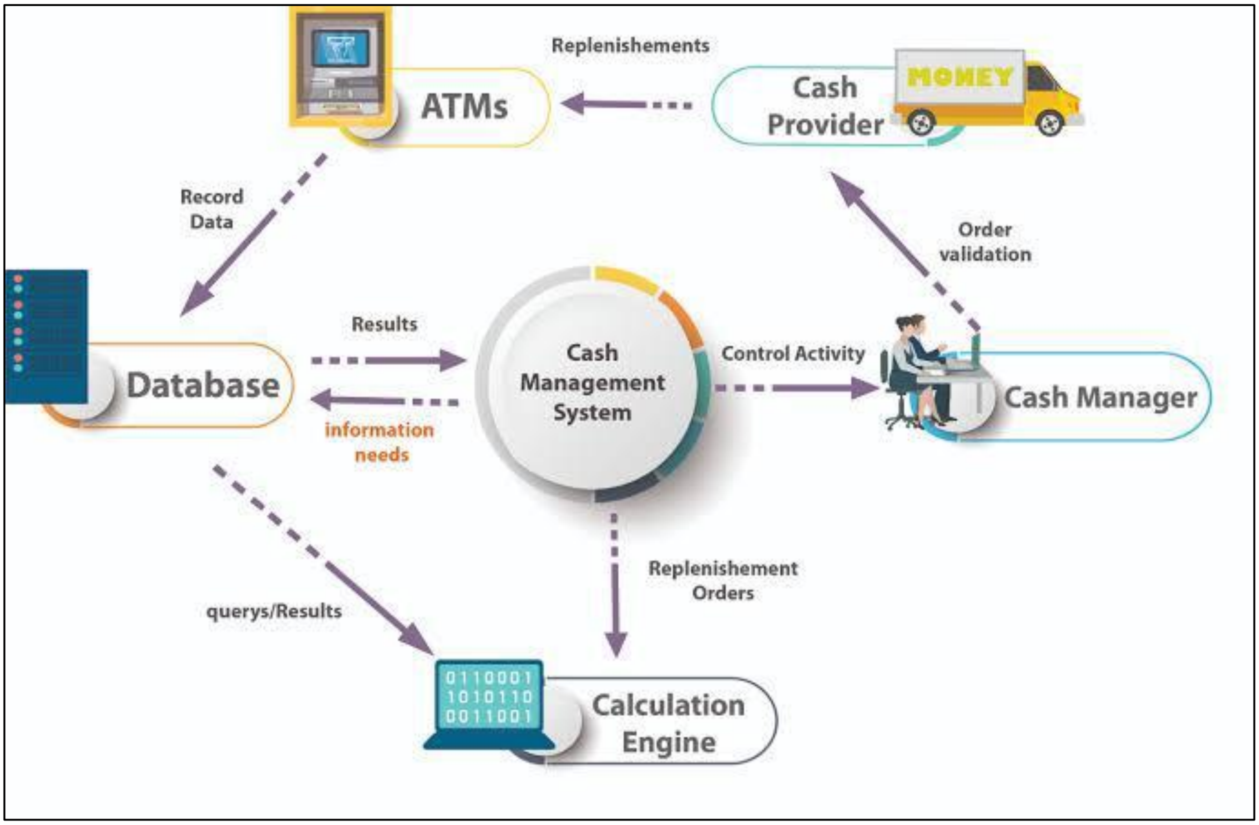
Activity diagrams represent the dynamics of system. In other words we can say, that it shows the workflow of a system. Activity Diagrams are like Flow Charts, but Flow Charts are usually limited to sequential activities while Activity Diagram can show parallel activates as well.



Activity diagram of our project

7.2.3. Flow Chart Diagram:

The Flow Chart shows the flow of procedure of whole system.



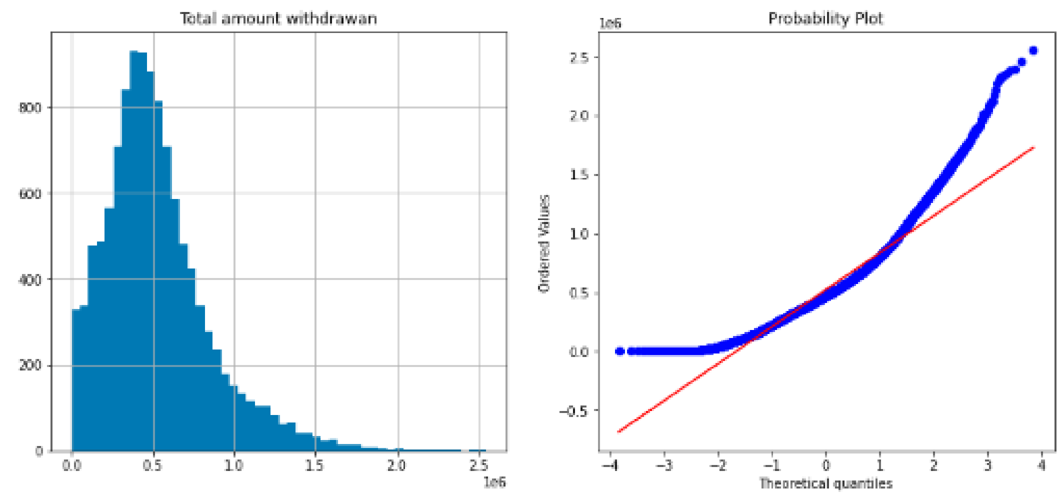
State chart Diagrams of our project:

CODE AND OUTPUT:

CODE AND OUTPUT for Dataset 1:

In [2]:
```
plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
data['Total amount Withdrawn'].hist(bins=50)
plt.title('Total amount withdrawan')
plt.subplot(1,2,2)
stats.probplot(data['Total amount Withdrawn'], plot=plt);
data['Total amount Withdrawn'].describe().T.round()
#data=pd.get_dummies(data,columns=['weekday','festival_religion','working_day','holiday_sequence'])
```

Out[2]:
```
count       11589.0
mean       522306.0
std        324817.0
min           100.0
25%        305700.0
50%        470000.0
75%        671600.0
max       2549800.0
Name: Total amount Withdrawn, dtype: float64
```



DATASET 1 VISUALIZATION

In [9]:
```
linear = LinearRegression()
linear.fit(xtrain, ytrain)
# make predictions
lin_pred = linear.predict(xtest)
# Prediction vs Actual
linpred = pd.DataFrame(lin_pred[-10:]) # predicting last 10 values
linpred.rename(columns = {0: 'predicted'}, inplace=True) # renaming the column
linpred = linpred.round(decimals=0) # rounding the decimal values
d = pd.DataFrame(data['Total amount Withdrawn']).tail(10) # calling last 10 values of original amt wothdrawn
d=d.rename({'Total amount Withdrawn':'Actual'}, axis=1)
linpred.index = d.index # mapping the index of both dataframe
d['ATM Name']=data['ATM Name'].tail(10)
linok = pd.concat([linpred, d], axis=1)
linok['accuracy'] = round(linok.apply(lambda row: row.predicted /row.Actual *100, axis = 1),2)
linok['accuracy'] = pd.Series(["{0:.2f}%".format(val) for val in linok['accuracy']],index = linok.index)
linpred
d
linok.index.names=['BANK_ID']
linok
# linok = linok.assign(day_of_week = lambda x: x.linok.index.data())
```

Out[9]:

BANK_ID	predicted	Actual	ATM Name	accuracy
11579	468600.0	468600	Big Street ATM	100.00%
11580	317400.0	317400	Mount Road ATM	100.00%
11581	424700.0	424700	Airport ATM	100.00%
11582	1154900.0	1154900	KK Nagar ATM	100.00%
11583	1120300.0	1120300	Christ College ATM	100.00%
11584	468800.0	468800	Big Street ATM	100.00%
11585	305100.0	305100	Mount Road ATM	100.00%
11586	709900.0	709900	Airport ATM	100.00%
11587	408700.0	408700	KK Nagar ATM	100.00%
11588	700400.0	700400	Christ College ATM	100.00%

LINEAR REGRESSION CODE

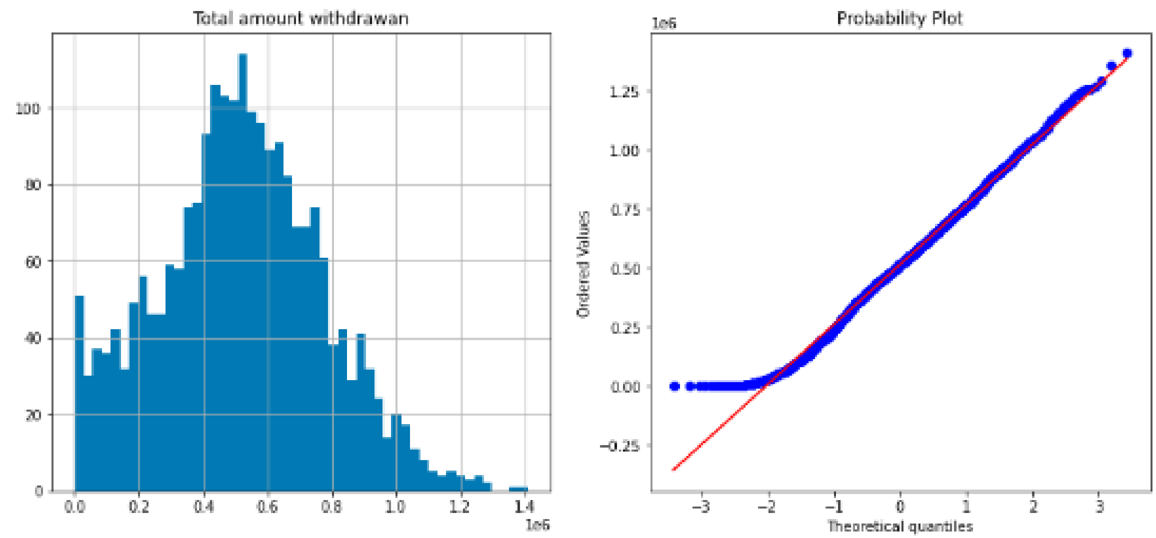
OUTPUT OF DATASET 1:

```
In [38]: from sklearn import metrics
         lin_test=ytest.mean()
         pred=lin_pred.mean()
         print(lin_test,pred)
         accuracy=pred/lin_test
         print(accuracy*100)

         426007.9810181191 426007.981018119
         99.99999999999999

In [39]: print('Variance:', linear.score(Xtest, ytest))

         Variance: 1.0

In [40]: MAE_lr=metrics.mean_absolute_error(ytest,lin_pred)
         MSE_lr=metrics.mean_squared_error(ytest,lin_pred)
         RMSE_lr=np.sqrt(MSE_lr)
         r2_lr=metrics.r2_score(ytest,lin_pred)

         print(MAE_lr)
         print(MSE_lr)
         print(RMSE_lr)
         print(r2_lr)

         2.4261731230124355e-10
         1.311851734153433e-19
         3.6219493841761967e-10
         1.0
```

OUTPUT

CODE AND OUTPUT for Dataset 2:

In [40]:
```
plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
df['total_amount_withdrawn'].hist(bins=50)
plt.title('Total amount withdrawan')
plt.subplot(1,2,2)
stats.probplot(df['total_amount_withdrawn'], plot=plt);
df['total_amount_withdrawn'].describe().T.round()
data=df
obj_df = df.select_dtypes(include=['object']).copy()
```



DATASET 2 VISUALIZATION

In [47]:
```
linear = LinearRegression()
linear.fit(Xtrain, ytrain)# make predictions
lin_pred = linear.predict(Xtest)# Prediction vs Actual
linpred = pd.DataFrame(lin_pred[-10:]) # predicting last 10 values
linpred.rename(columns = {0: 'lin_predicted'}, inplace=True) # renaming the column
linpred = linpred.round(decimals=0) # rounding the decimal values
d = pd.DataFrame(df['total_amount_withdrawn']).tail(10)# calling last 10 values of original amt wothdrawn
d['weekday']=df['weekday'].tail(10)
linpred.index = d.index # mapping the index of both dataframe
linok = pd.concat([linpred, d], axis=1)
linok['accuracy'] = round(linok.apply(lambda row: row.lin_predicted /row.total_amount_withdrawn *100, axis = 1),2)
linok['accuracy'] = pd.Series(["{0:.2f}%".format(val) for val in linok['accuracy']],index = linok.index)
linok
```

Out[47]:

	lin_predicted	total_amount_withdrawn	weekday	accuracy
2234	508440.0	308400	WEDNESDAY	164.86%
2235	488103.0	312600	THURSDAY	156.14%
2236	461633.0	337100	FRIDAY	136.94%
2237	572903.0	251100	SATURDAY	228.16%
2238	374079.0	182700	SUNDAY	204.75%
2239	530596.0	447400	MONDAY	118.60%
2240	506257.0	153800	TUESDAY	329.17%
2241	518418.0	167100	WEDNESDAY	310.24%
2242	524031.0	317400	THURSDAY	165.10%
2243	209299.0	305100	FRIDAY	68.60%

In [54]:
```
from sklearn import metrics
lin_test=ytest.mean()
lin_pred=pred.mean()
print(lin_test,lin_pred)
accuracy=lin_test/lin_pred
print(accuracy*100)
```

```
246600.89086859688 480655.85175691283
51.3050844938663
```

LINEAR REGRESSION CODE AND OUT PUT

```
In [50]: import xgboost as xgb

         model_xgb=xgb.XGBRegressor()
         model_xgb.fit(Xtrain,ytrain)
         xgb_pred=model_xgb.predict(Xtest)

In [51]: # Prediction vs Actual
         linpred = pd.DataFrame(xgb_pred[-10:]) # predicting last 10 values
         linpred.rename(columns = {0: 'lin_predicted'}, inplace=True) # renaming the column
         linpred = linpred.round(decimals=0) # rounding the decimal values
         d = pd.DataFrame(df['total_amount_withdrawn']).tail(10)# calling last 10 values of original amt wothdrawn
         d['weekday']=df['weekday'].tail(10)
         linpred.index = d.index # mapping the index of both dataframe
         linok = pd.concat([linpred, d], axis=1)
         linok['accuracy'] = round(linok.apply(lambda row: row.lin_predicted /row.total_amount_withdrawn *100, axis = 1),2)
         linok['accuracy'] = pd.Series(["{0:.2f}%".format(val) for val in linok['accuracy']],index = linok.index)
         linok
```

Out[51]:

	lin_predicted	total_amount_withdrawn	weekday	accuracy
2234	29213.0	308400	WEDNESDAY	9.47%
2235	356450.0	312600	THURSDAY	114.03%
2236	197966.0	337100	FRIDAY	58.73%
2237	90477.0	251100	SATURDAY	36.03%
2238	267407.0	182700	SUNDAY	146.36%
2239	199052.0	447400	MONDAY	44.49%
2240	591902.0	153800	TUESDAY	384.85%
2241	127836.0	167100	WEDNESDAY	76.50%
2242	250156.0	317400	THURSDAY	78.81%
2243	94842.0	305100	FRIDAY	31.09%

```
In [53]: lin_test=ytest.mean()
         xl_pred=xgb_pred.mean()
         print(lin_test,xl_pred)
         accuracy=lin_test/xl_pred
         print(accuracy*100)

         246600.89086859688 286123.56
         86.18685183209854
```

XG BOOST CODE AND OUT PUT

MOTIVATION:

The motivation for doing this project was to be familiar with Artificial Intelligence, Data Science and, Machine Learning's core concepts and physical implementations.

TASK DISTRIBUTION:

Muhammad Owais Mushtaq:
Final Report + Proposal + Dataset + Solved Linear Regression queries
Muhammad Usman Umar:
Worked on Data set 1 and 2 applied Linear Regression and XG Boost
Shaharyar Amjad:
Presentation + Worked on Data set 1 applied Linear Regression

CONCLUSION:

Working on two different datasets we found 100% accuracy in result of dataset1 (Aggregated dataset) through linear regression however in other dataset we found 50% accuracy of prediction from linear regression. Hence, we then used XG Boost to increase our accuracy till 85 %.