IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 3, March 2020

Data Analytics and ML: Bank Transactions over a Long Period of Time

Kavya K[1], Manish Y M[2], Priyanka P A[3], Savinay Shukla[4]

BE, Department of ISE, BNMIT, Bangalore, India [1,2]

Student, MBA, CMS Business School, Bangalore, India [3]

BTech, Department of IT, Manipal University, Jaipur, India [4]

Abstract: Banks have been the most important institutions of money lending and deposits. Primary functions include accepting deposits, offering loans, credit, overdraft, providing liquidity and discounting of bills. Secondary functions include providing safe custody of valuables, loans on valuables, corporate and consumer finances. Though the structure of banks has remained the same, the functionalities have been boosted. Automated tools, bots and computers have modernized the banking system. The dataset accumulated over a period of time is so huge that, automation tools and computer programs are the need of the day. In this paper we have tried to enhance the present bank credit-debit system by the use of Artificial Intelligence. Machine learning is a subset of AI and directly trains the machine by feeding the historic and runtime data collected during transactions. The machine which is trained is now capable of taking decisions, thereby making predictions. This would characterize the dataset as stored and predicted outcomes. Every business enthusiast would have keen interest to carefully study the performance of a financial institute for his/her benefit. In this assignment we have used both classification and regression algorithms to create a ML model of prediction. Linear regression model is designed from scratch using formula method. Classification algorithms like Support Vector Machine (SVM), Random Forest Classifier and KNN algorithms are effectively applied to fit to the dataset. Comparisons must be made during implementation to understand the pattern of predicted data. Regression algorithms like linear regression (developed from scratch) will be a boost to the accuracy of the assignment (categorical data excluded).

Keywords: accepting deposits, offering loans, credit, overdraft, providing liquidity and discounting of bills, Automated tools, bots and computers, Machine learning, Support Vector Machine (SVM), Random Forest Classifier and KNN algorithms, linear regression (developed from scratch), historic and runtime data collected during transactions, AI.

I. INTRODUCTION

In this information era, huge amount of data is being stored, exchanged and conditioned. The volume of data that one has to deal with has exploded to unimaginable levels. Most of the data exists in its crude form and needs to be converted to useful format before analysis. This process of converting raw data into useful format is called data pre-processing. Real world data is [1]

- Incomplete: consists of missing attribute values or consists of only aggregate data.
- Noisy: containing errors or outliers.
- Inconsistent: containing discrepancies in code.

Our assignment consists of the following attributes. Table 1 shows the attributes with their description

Sl.no	Attributes	Description
1.	ID	ID given to Borrower.
2.	loan_amnt	amount of money requested by the borrower.
3.	funded_amnt	The total amount committed to that loan at that point in time.
4.	funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
5.	term_months	Time in months given to the borrower to return the loan amount.
6.	int_rate	The interest rate of the loan, as a proportion (a rate of 11% would be stored as 0.11).
7.	installment	The monthly installments owed by the borrower if the loan is funded.
8.	emp_length	Time in years where the borrower has been employed. (0.5 = < 1 year 1= 1 year

 DOI 10.17148/IJARCCE.2020.9331

		2= 2 years 3= 3 years 4= 4 years 5= 5 years 6= 6 years 7= 7 years 8= 8 years 9= 9 years 10= 10+ years)
9.	home_ownership	State or fact of exclusive rights and control over borrower property. 1= mortgage 2= rent 3= own
10.	annual_inc	The annual Income of the borrower.
11.	verification_status	A Verification Stage is a three-stage indicator of the progress on the loan, based on the status of verification of the borrower's identity, information and documentation. 0= Not Verified 1= Source Verified 2= Verified
12.	issue _year	The year in which the loan is issued.
13.	loan_status	• 0-Fully Paid (A 'fully paid' loan has been repaid in full including all principal and interest payments.) • 1- Charged off(An amount of debt that is unlikely to be collected) or Default
14.	purpose	The main reason the borrower applies for a loan. 1= debt_consolidation 2= credit_card 3= car 4= vacation 5= home_improvemnt 6= small_busineses 7= major_purchase 8= medical
15.	earliest_cr_line_date	The month the borrower's earliest reported credit line was opened
16.	delinq_2yrs	The number of times the borrower had been 30+ days past due on a payment in the past 2 years.
17.	inq_last_6mths	The borrower's number of inquiries by creditors in the last 6 months.
18.	open_acc	Number of open credit lines on the borrower's credit file. (A line of credit (LOC) is a preset borrowing limit that can be used at any time.)
19.	pub_rec	The borrower's number of derogatory public records(bankruptcy filings, tax liens, or judgments).
20.	revol_bal	The borrower's revolving balance (amount unpaid at the end of the credit card billing cycle).
21.	revol_util	The borrower's revolving line utilization rate (the amount of the credit line used relative to total credit available).
22.	total_acc	Total number of historical credit lines on the borrower's file.

23.	out_prncp	Remaining outstanding principal for total amount funded.
24.	out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
25.	total_pymnt	Payments received to date for total amount funded.
26.	total_pymnt_inv	Payments received to date for portion of total amount funded by investors.
27.	last_pymnt_d	Last month payment was received.
28.	total_rec_late_fee	Late fees received to date.
29.	last_pymnt_amnt	Last total payment amount received.
30.	application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers. 1= individual 2= joint app
31.	tot_coll_amt	Total collection amounts ever owed
32.	tot_cur_bal	Total current balance of all accounts
33.	open_act_il	Number of currently active installment trades.
34.	open_il_24m	Number of installment accounts opened in past 24 months.
35.	mths_since_rcnt_il	Months since most recent installment accounts opened.
36.	il_util	Ratio of total current balance to high credit/credit limit on all install acct.
37.	acc_open_past_24mths	Number of trades opened in past 24 months.
38.	avg_cur_bal	Average current balance of all accounts.
39.	bc_open_to_buy	Total open to buy on revolving bankcards.
40.	bc_util	Ratio of total current balance to high credit/ credit limit for all bankcard account.
41.	chargeoff_within_12_mths	Number of charge-offs within 12 Months.
42.	delinq_amnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
43.	mths_since_recent_bc	Months since most recent bankcard account opened.
44.	mths_since_recent_inq	Months since most recent inquiry.
45.	num_accts_ever_120_pd	Number of accounts ever 120 or more days past due.
46.	num_actv_bc_tl	Number of currently active bankcard accounts.
47.	num_actv_rev_tl	Number of currently active revolving trades.
48.	num_bc_sats	Number of satisfactory bankcard accounts.
49.	num_bc_tl	Number of bankcard accounts.
50.	num_il_tl	Number of Installment accounts.
51.	num_op_rev_tl	Number of open revolving accounts.
52.	num_rev_accts	Number of revolving accounts.
53.	num_rev_tl_bal_gt_0	Number of revolving trades with balance >0
54.	num_sats	Number of satisfactory accounts.
55.	num_tl_op_past_12m	Number of accounts opened in the last 12 Months.
56.	pct_tl_nvr_dlq	Percent of trades never deliquent
57.	percent_bc_gt_75	Percentage of all bankcard accounts>75% of limit.
58.	pub_rec_bankruptcies	Number of public record bankruptcies.
59.	tot_hi_cred_lim	Total high credit /credit limit.
60.	total_bal_ex_mort	Total credit balance excluding mortgage.
61.	total_bc_limit	Total bankcard high credit /credit limit.
62.	total_il_high_credit_limit	Total installment high credit /credit limit.

Table 1

II.PROBLEM STATEMENT

In a bank huge dataset is produced with everyday transaction and with ever increasing deposits, loans, insurance policies, over drafts and other services. A bank with huge customers is considered and transaction data of 20 years has been recorded. The identity of the customer is morphed. Unique IDs to be presented to the same. Data needs to be examined for pattern recognition and data pre-processing needs to be carried out to –

- Fill the missing values or null values
- Remove redundant entries.
- Treat NaN values.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 3, March 2020

- Replace string values with their numerical counterparts.
- Create a sketch of post-assigned categorical values in each column defining a particular attribute.

The pre-processed data needs to be fed to the machine for training. The patterns would train the machine to make predictions in all possible situations. Classification algorithms like SVM, Random forest classifier, KNN and logistic regression need to be applied. Linear regression is modelled from scratch without using libraries for more accuracy and F1 score.

III. METHODOLOGY

i. Data acquisition – Data acquisition is carried out. Everyday transactions are recorded and stored in the database. Figure 1 shows the process of data acquisition. 20 years data transaction consists of about 1.5 million unique transaction IDs. About 140 parameters or attributes are part of this dataset. Few of them have been tabulated above. [1] [2]

ii. Data Inspection – The acquired data is inspected before data pre-processing. The data needs to be preprocessed before analytics or training. Figure 2 shows the process of data inspection. [3]

iii. Data Visualization – Graphical analysis of the dataset which is huge in nature is essential. Figure 3 shows a bar graph of verification_status v/s count. Figure 4 shows a count plot of loan purpose. Figure 5 shows a hue plot of home_ownership against loan_status. [4]

iv. Correlation is carried out and heat map plotted as shown in figure 6. Regions of strong and weak correlation is described by the color bar. Neutral values are ignored.

v. Linear regression is used to create a ML model for columns with non-categorical behavior. Figure 7 shows the code bit of the same using formula method (no libraries used). [5]

vi. Classification algorithms like SVM, KNN and Random forest classifier are applied to the model.

vii. The predicted values are well tabulated, accuracy measured and compared. [6]

viii. The predictions are sent back to be stored in the database for a closed loop execution in the coming years and will be continuously compared with the then run time values or transactions.

ix. The master table or the data table with raw or crude data is updated each time a new transaction takes place and the manipulated dataset is treated with time before execution.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.linear_model import LogisticRegression
        from sklearn.model_selection import train_test_split

In [2]: df= pd.read_csv(r'C:\Users\nups0\Desktop\dataset.csv')

In [3]: df
Out[3]:
```

	id	loan_amnt	funded_amnt	funded_amnt_inv	term_months	int_rate	installment	emp_length	home_ownership	annual_inc	...	num_tl_90g_dpd_
0	1474286	30000	30000	30000	36	22.35	1151.16	5.0	1	100000.0	...	
1	1474287	40000	40000	40000	60	16.14	975.71	0.5	1	45000.0	...	
2	1474288	20000	20000	20000	36	7.56	622.68	10.0	1	100000.0	...	

Figure 1 shows the process of data acquisition.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 3, March 2020

```
In [6]: df.size
Out[6]: 614481

In [7]: df.shape
Out[7]: (7063, 87)

In [8]: df.dtypes
Out[8]: id                           int64
        loan_amnt                    int64
        funded_amnt                  int64
        funded_amnt_inv              int64
        term_months                  int64
                                     ...
        tot_hi_cred_lim              int64
        total_bal_ex_mort            int64
        total_bc_limit               int64
        total_il_high_credit_limit   int64
        disbursement_method          int64
        Length: 87, dtype: object
```

Figure 2 shows the process of data inspection.

```
In [14]: f= plt.subplots(figsize=(15,5))
         color_types=['#00FF00','#00FFFF','#FFAAEE']
         sns.countplot(x="verification_status",palette=color_types, data=df).set_title("Verification status graph")

Out[14]: Text(0.5, 1.0, 'Verification status graph')
```

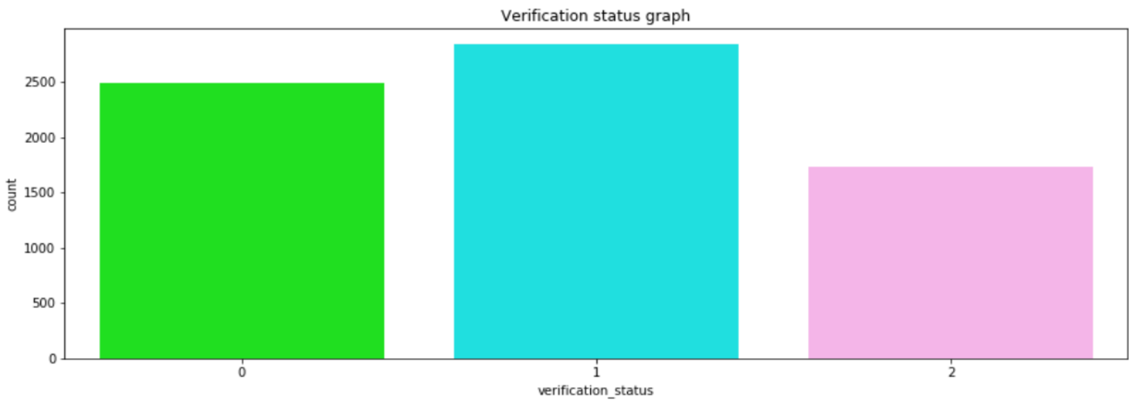

Figure 3 shows a bar graph of verification_status v/s count.

```
In [17]: f= plt.subplots(figsize=(15,5))
         color_types=['#00FF00','#00FFFF','#FFAAEE']
         sns.countplot(x="purpose",palette=color_types, data=df).set_title("Loan purpose graph")

Out[17]: Text(0.5, 1.0, 'Loan purpose graph')
```

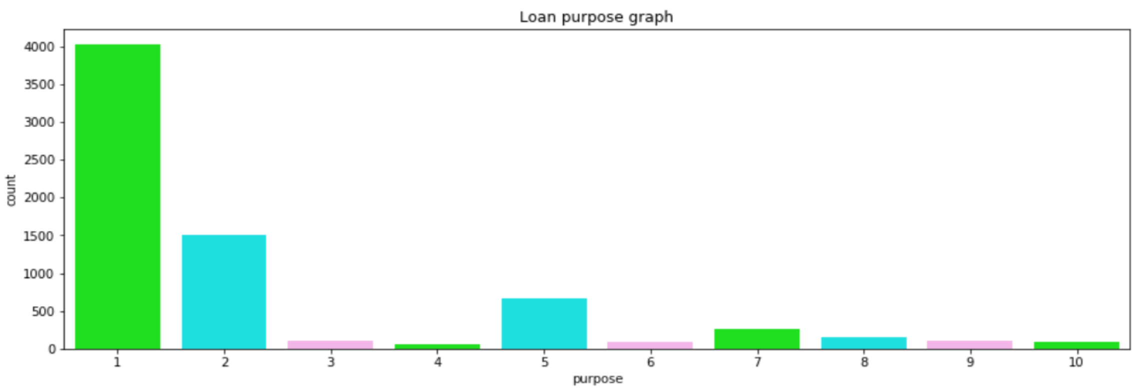

Figure 4 shows a count plot of loan purpose.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 3, March 2020

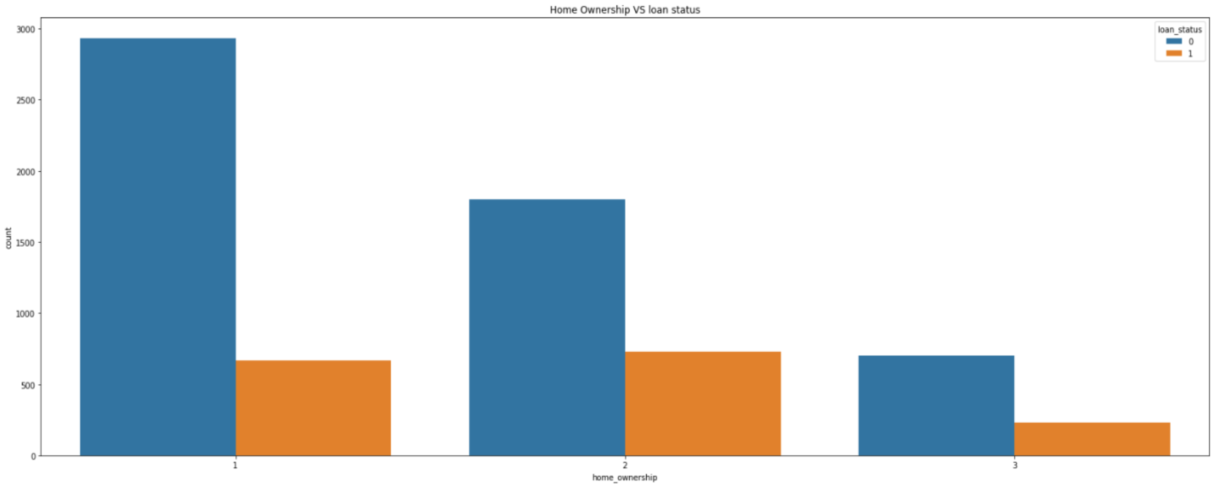

Figure 5 shows a hue plot of home_ownership against loan_status.

	id	loan_amnt	funded_amnt	funded_amnt_inv	term_months	int_rate	installment	emp_length	home_ownership	annual_inc	...
id	1.000000	0.027341	0.027341	0.027288	0.041211	0.130415	0.044465	-0.035195	0.057782	-0.055215	...
loan_amnt	0.027341	1.000000	1.000000	0.999995	0.390274	0.106911	0.951471	0.082418	-0.089565	0.324420	...
funded_amnt	0.027341	1.000000	1.000000	0.999995	0.390274	0.106911	0.951471	0.082418	-0.089565	0.324420	...
funded_amnt_inv	0.027288	0.999995	0.999995	1.000000	0.390464	0.107005	0.951404	0.082396	-0.089583	0.324391	...
term_months	0.041211	0.390274	0.390274	0.390464	1.000000	0.383300	0.167894	0.041518	-0.065941	0.041844	...
...	...	...	...	...	...	...	...	...	...	...	...
tot_hi_cred_lim	-0.098492	0.279343	0.279343	0.279400	0.079582	-0.112946	0.250768	0.139870	-0.368591	0.557846	...
total_bal_ex_mort	-0.032480	0.217649	0.217649	0.217665	0.084091	0.058044	0.213551	0.046978	-0.127184	0.399346	...
total_bc_limit	-0.109086	0.311789	0.311789	0.311762	0.045502	-0.234596	0.277310	0.066914	-0.062696	0.348812	...
total_il_high_credit_limit	-0.039759	0.159554	0.159554	0.159590	0.064376	0.032922	0.155136	0.048567	-0.113876	0.360716	...
disbursement_method	0.127683	-0.033662	-0.033662	-0.033713	-0.014596	0.116828	-0.011868	0.002189	0.013929	-0.005046	...

Figure 6 shows the heatmap coefficients.

```
In [12]: #mean of X and Y
         mean_x = np.mean(X)
         mean_y = np.mean(Y)
         #total number of values
         m = len(X)
         #formula to calculate b1 and b0
         numer= 0
         denom= 0
         for i in range(m):
             numer+= (X[i] - mean_x) * (Y[i] - mean_y)
             denom+= (X[i] - mean_x) ** 2
         b1 = numer/denom
         b0 = mean_y - (b1 * mean_x)
         #b1 and b0 are m and c respectively in y=mx+c
         print(b1,b0)

         0.08346747763810816 12009.934165736577
```

Figure 7 shows the code bit of the same using formula method (no libraries used).

IV.RESULTS

- Figure 8 shows the accuracy comparison of the classification algorithms. Random Forest Classifier model has an accuracy of 99.9%, Logistic regression model has an accuracy of 99.75%, KNN model has an accuracy of 80.89% at K=3 and 5 and SVM model has an accuracy of 75.87%. So, we Random Forest and Logistic Regression method show very good accuracies and are a very good fit to this assignment.
- Linear Regression model shows an accuracy of 91.155%. Figure 9 shows the r^2 value of the linear regression model.
- Figure 10 shows the predictions or the X_test values after linear regression.

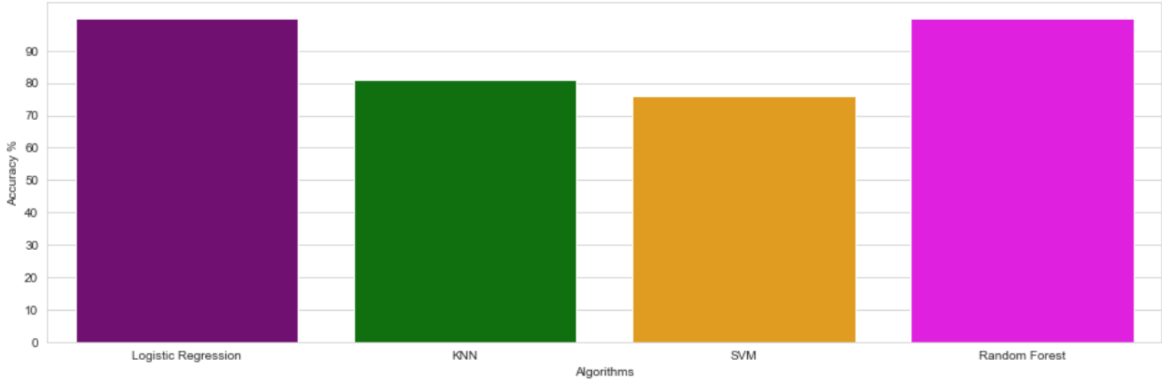

Figure 8 shows the accuracy comparison of the classification algorithms.

```
In [14]: ss_t = 0
         ss_r = 0
         for i in range(m):
             y_pred = b0 + b1 * X[i]
             ss_t += (Y[i] - mean_y) ** 2
             ss_r += (Y[i] - y_pred) ** 2
         r2 = 1 - (ss_r/ss_t)
         R2=r2*10
         print(R2*100)

         91.15564927258013
```

Figure 9 shows the r^2 value of the linear regression model.

```
In [25]: Y[250:500]

Out[25]: array([10000,  5000, 10000, 10000, 20000,  3500,  9500, 30000, 15000,
                15000,  5000, 25000, 10800, 40000,  5500, 10000, 10000, 16800,
                 3000, 10000,  7000,  6000,  8000, 30000, 12000, 15500, 25000,
                 1300, 25000,  5000, 10000,  6400, 11000, 28000, 12000, 12350,
                35000, 30000, 30000,  1000, 10000, 11000, 35000,  7000, 31400,
                16000,  3000, 10000,  1600, 17000,  5500,  8000,  2500, 15000,
                18000, 35000, 15000, 10000,  6500, 15000, 24000, 10000, 25000,
                29700, 37000, 16000, 19050,  9000, 24000, 35000, 10000, 10000,
                40000, 18000, 16000, 15000,  3200, 40000, 40000,  1000,  2500,
                 8000, 19000,  9000, 12000, 30000, 10000, 25000, 16000,  6000,
                12000, 30000,  6500, 27650, 24000, 20000, 25000, 25000, 10000,
                35500, 20000,  3000,  6000, 28000, 20000, 14500, 25000, 10000,
                 7500, 30000,  2850,  2000, 28000, 24000, 40000, 10000,  8800,
                19000,  6125, 20000, 40000, 19975, 21000, 23500,  3900, 16500,
                15000,  8500, 32425, 30000,  4000,  8000, 30000, 15000, 15000,
                 7200,  4800, 20000, 18000,  5000, 27000, 10000,  5000, 10000,
                 9500, 28800, 31200,  5000,  1400, 12000,  5000,  9000, 20550,
                 7000,  2000, 15600,  4500, 40000, 21000, 16000, 10000, 20000,
                 8000, 10000,  1500, 40000,  6600, 17000, 11000,  4800, 28000,
                12000,  4500,  2600,  5000,  6475,  4000,  2000,  3600, 10000,
                18000,  4000, 10000,  2000, 10000,  3000,  2000, 10000, 15000,
                30000,  4500,  3200,  7500, 35000, 10400,  6300,  8000, 34000,
                30000, 20000, 30000, 15000, 19200,  8000, 40000, 11200, 10000,
                 5500, 12000, 21000,  6000,  8500,  2000, 40000, 12000, 25000,
                13500, 10000, 40000,  3000, 12000, 23500, 10000, 11000, 22000,
                11775, 17000,  6000, 38000, 40000, 30000,  7200,  9600, 15000,
                10000,  5000,  6025,  9500,  9000,  8000, 15000,  3000,  5000,
                 3000,  8000, 12000, 16000, 14000,  4000,  4800], dtype=int64)
```

Figure 10 shows the predictions or the X_test values after linear regression.

DOI 10.17148/IJARCCE.2020.9331

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 3, March 2020

V. CONCLUSIONS

A Bank proactive in business in this 21st century world has many day to day transactions. Data analytics had to be carried out on the data –both historical and present trend to draw inference. The goal was to create or improve the ML model and carry out accuracy check comparison. A python code was written and executed in the Jupyter platform to analyse and draw conclusions. Classification algorithms like Support Vector Machine (SVM), Random Forest Classifier and KNN algorithms are effectively applied to fit to the dataset. Comparisons must be made during implementation to understand the pattern of predicted data. Random Forest Classifier model has an accuracy of 99.9%, Logistic regression model has an accuracy of 99.75%, KNN model has an accuracy of 80.89% at K=3 and 5 and SVM model has an accuracy of 75.87%. We can conclude that Random Forest Classifier and Logistic Regression models are the best fit to this dataset. Since this data also behaves well for Linear regression algorithm, Linear regression is modelled from scratch without using libraries for more accuracy (91.155%) and F1 score.

REFERENCES

[1]. Principles of data mining- DJ Hand - Drug safety, 2007 - Springer

[2]. The Python Standard Library — Python 3.7.1rc2 documentation-https://docs.python.org/3/library/

[3]. Research on Data Preprocess in Data Mining and Its Application- J Zhi-gang, JIN Xu - Application Research of Computers, 2004 - en.cnki.com.cn

[4]. Data Mining and Analytics: A Proactive Model - http://www.ijarcce.com/upload/2017/february-17/IJARCCE%20117.pdf

[5]. A comparative analysis on linear regression and support vector regression- DOI: 10.1109/GET.2016.7916627

[6]. Data Warehousing Architecture and Pre-Processing- Vishesh S, Manu Srinath, Akshatha C Kumar, Nandan A.S.- IJARCCE, vol 6, issue 5, May 2017.