Predictive Modeling and Analysis of Logistic Regression and k-Nearest Neighbor for Personal Loan Campaign

Bhavya Alankar[1], Iftikhar Alam[2]

{bhavya.alankar@gmail.com[1], iftikharalam97@gmail.com[2]}

School of Engineering Science & Technology, Jamia Hamdard, New Delhi-110062, India[1] , School of Engineering Science & Technology, Jamia Hamdard, New Delhi-110062, India[2]

Abstract. Science is the knowledge which a person understands and that can be taught to a computer. Extensive research has been made to develop appropriate machine learning algorithms for different classification or function approximation problems. Some of the machine learning methods depends on the characteristics of the data set and the requirements of the business domain. This case study provides the predictive performance of different classification methods for identifying the potential customers who have a higher probability of purchasing the loan. For this we will build two statistical model, a logistic regression model and a k-nearest neighbor model. This study works on the following two techniques to build the model and provides a guideline for similar comparison studies and to find the best one out.

Keywords: Personal loan management, data science, classification, machine learning techniques, logistic regression, k-Nearest Neighbor.

1 Introduction

Some of the existing researches focus on the achievable classification accuracy of different machine learning algorithms and techniques [1]. Machine learning algorithms are used in many of the software programs, such as Microsoft's paperclip in Office, spam filters, voice recognition software etc. They are also some of the set of algorithms that decide whether a bank will give you a loan or not. Having a wide variety of available algorithms and the volume of data that is to be analyzed, the selection of the right technique to use for a new problem is an important task, and there can be some differences in the predictions of the resulting models [2]. Data analysis is not easy, and the problem is that only some people can explain how to do it. Data analysts have many tools at their hands, from linear regression to classification trees and even deep learning, and these tools have all been in use and taught to computers[3]. Recently, many algorithms for ordinal regression have been used and proposed from a machine learning perspective. And since binary classification is much better than ordinal regression, a general framework to systematically reduce the latter to the former[4].

So for the same problem, how to pick one tool out of tools that are available, again there is a logical way in which we can do this, we can make some assumptions about the underlying data distribution and model types and then choose a tool that is relevant for those assumptions and then see whether the tool works. Hence, we are going to use a couple of tools and then run those

ICIDSSD 2020, February 27-28, New Delhi, India

DOI 10.4108/eai.27-2-2020.2303232

tools on the following data set and then basically we are going to judge which tool is better based on just outcomes in terms of the prediction error or the confusion matrix and so on.
There are many types of machine learning that we can use for solving the problem, but here we are going to use supervised learning. Supervised classification algorithms are used to produce a learning model from a training or data set. Various successful techniques have been proposed to solve the problem in the binary classification case [5]. There is a set of data that consists of a set of input data that is the target data. This is usually written as a set of data (x_i, t_i), where the inputs are x_i, the targets are t_i, indexed by i running from 1 to some upper limit N [6].So we are going to look at how well is it going to do on the test data and then simply pick the best method in terms of the results on the test data.

2 Related Works

Classification occurs and finds use in a wide range and it could cover any context in which some decision is made based on data. In statistics, linear discriminant analysis and logistic regression have made a major method for classification problems. The results of previous studies are often in direct contradiction, with some authors claiming that decision trees and kNN method are superior to logistic regressions, but this is not always the case. Logistic Regression may give a better result and accuracy in some of the cases. For example, Mingers (Mingers 1987) compared the ID3 rule induction algorithm to multiple regression. But the results of this comparison suffer from several limitations. Gilpin et al. (Gilpin and Ohlsen 1990) compared regression trees, stepwise linear discriminant analysis, logistic regression, predicting the probability of one-year survival of patients who had myocardial infarctions. The methods were compared as to their sensitivity and specificity. There were no significant differences founded between any of these methods. The European stat log project (Michie et al. 1994) was a study comparing around 20 methods on 20 datasets. The purpose was to classify the performance of the classification techniques based on the problem types. This attempt had only limited success due to the wide variability of the datasets. So, for classification problems, it is better to conduct a comparison study and study the performance of different methods using different parameters available such as accuracy, confusion matrix, R-square method etc., in order to identify relationships among the predictions of different classification methods.

3 Problem Description

This case is about a bank which has a growing customer base. Majority of these customers are depositors. The number of borrowers is quite small, and the bank is interested in expanding their base to bring more loan business in the process, earn more through the interest on loans. The management wants to explore some ways to convert their customers to personal loan customers. The department wants to build a model that will help them to identify those potential customers who have a higher probability of purchasing the loan so that it will increase their success ratio while at the same time reduce their cost of campaign.

The data set includes 5000 observations with fourteen variables divided into four different measurement categories. The binary category has five variables, including the target variable

personal loan, securities account, CD account, online banking, and credit card. The interval category contains five variables including age, experience, income CC avg and mortgage. The ordinal category includes the variables family and education. The last category is nominal with ID and Zip code. The variable ID does not add any interesting information, so it will be neglected in the problem.[12]

4 Methodology

There are number of methods which can be applied to classification problems, but which one to select is a difficult question. There is very rational way of choosing techniques for different problems but that means we must have some information, some knowledge about the underlying data, so that we can choose one. One standard thing everyone does nowadays is to take a number of techniques and apply all of them on the same problem and simply pick the technique that does the best right, so that's a very practical and utilitarian viewpoint of data science or data analysis algorithms and it works in many cases. The upside of this is that we don't have to really know much about what the algorithms even does, we just have to know if we have a binary classification problem and there is a laundry list of techniques that we can use, so if we have understanding at that level, then we can start using it. The downside is subtle and in most cases it might not be a problem, but in some cases it can be a serious problem which is that if the data you know is biased in terms of how the sampling has been done and it is not truly representative of what is happening then sometimes you might have a technique which works really well with the this data but when the new data comes in it, it might do very poorly, so always it's a good idea to understand the data and then kind of pick a technique but in this case, we are going to pick all of these techniques(Logistic Regression and k-Nearest Neighbor) and then run them. Then we will evaluate performance metrics, how well it does on the test data, it's a good performance metric to check. And then simply pick the best method in terms of results on the test data.

We will work on two methods for the following problem and that is logistic regression and the kNN method . The two techniques implementations are very different from each other for a classification problem. In further study, we will see the working of each of the two technique and choose the one that stand out the best.

4.1 Logistic Regression

Logistic regression is a machine learning technique and a classification algorithm that is used to predict the probability of a categorical dependent variable. Like all regression analysis, the logistic regression is a predictive analysis. Sometimes logistic regressions are difficult to interpret. Several logistic regression models used for analyzing have been developed, although it is not widely used [7]. Regression analysis is a process to find the relationship between dependent variable and independent variable. For example, let's take a variable y which is linearly dependent on the variable x, then regression analysis is used to find the constants a and b in the equation $y = ax + b$ that shows the linear relationship between the variables y and x [8]. Logistic regression is used to analyze data and to explain the relationship between one dependent variable and one or more independent variables. It also has some applications in testing hypotheses about relationships between a categorical outcome variable and one or more

categorical or continuous variables [9]. Using the logistic regression, we will build a classifier model based on the available data set.

4.2 k-Nearest Neighbor

k-Nearest Neighbor is also a machine learning technique and an algorithm that is used for classification and regression. K nearest neighbor's algorithm works by storing all variables cases and classifying them based on a similarity measure. kNN has applications in statistical estimation and pattern recognition since 1970's. The kNN is a simple but effective method for classification. The only drawbacks of kNN algorithm is its low efficiency and its dependency on the selection of a good value for k [10]. Given a test document d_t, the kNN algorithm find its k nearest neighbor among the training documents which forms the neighborhood of d_t [11]. As kNN belongs to the supervised learning domain, it has many applications in data mining, pattern recognition and intrusion detection.

Comparison of methods in Campaign for Bank Personal Loan

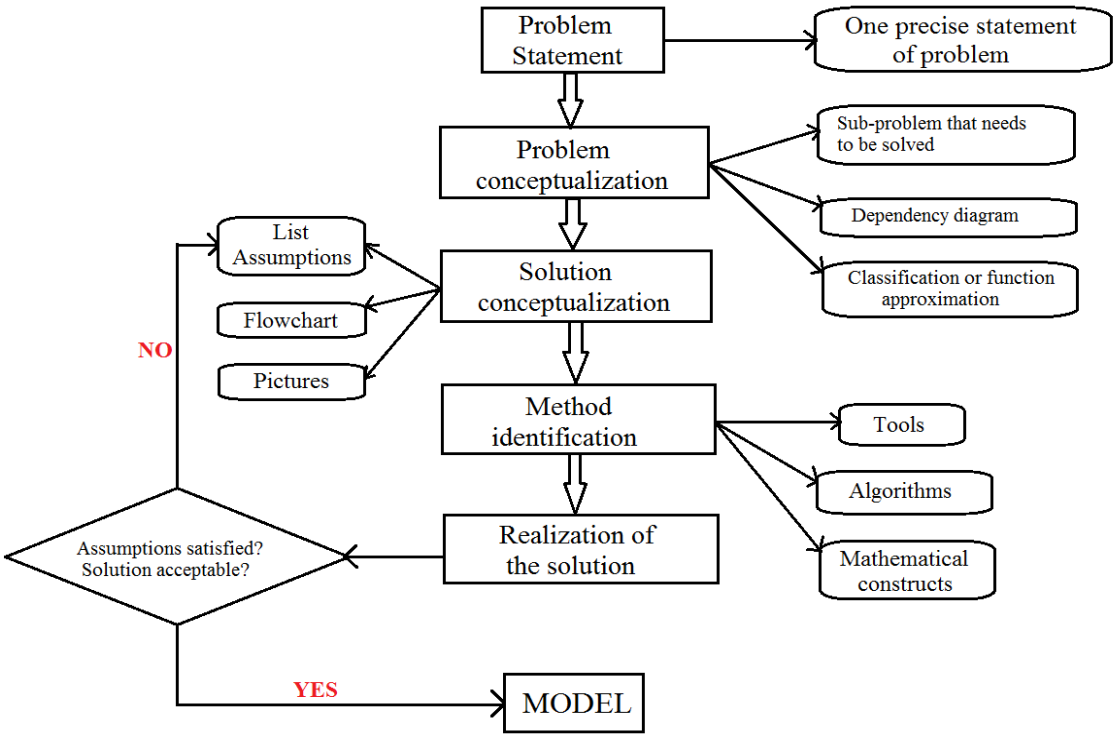


Fig. 1. Data Analytics framework

We will be given a binary classification problem with a problem statement. In many cases, we are presented with a problem, it is really not obvious exactly what data science problem we need to solve, so in those cases we have to think about some notion of problem conceptualization breaking down a very loosely worded problem statements and then under identifying these smaller problem statement as what type of problem they are either classification or function approximation problems and once we are able to solve the smaller problems, how do we put them in some logical arrangement of solution, so that we can solve the larger loosely frame problem, so that is actually where the intellectual exercise of data science comes in.

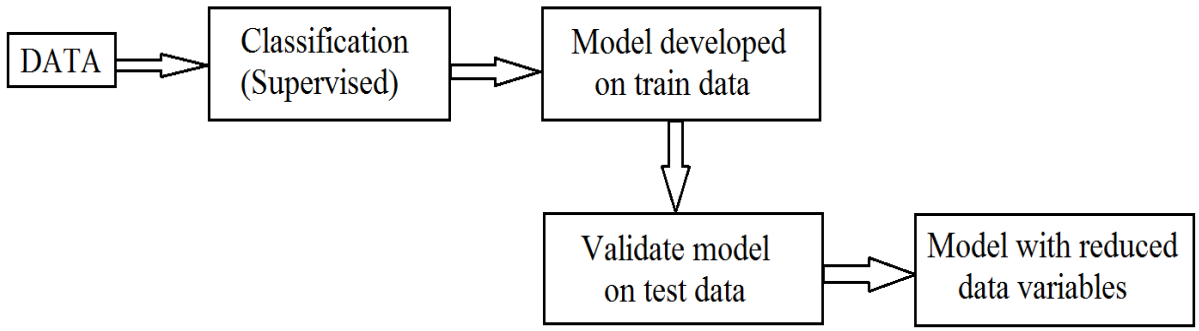


Fig. 2. Flowchart of building the models

If we put all the data into a flowchart based on the extra wrinkle that we threw into the problem in trying to reduce the number of input variables that we need to solve the problem. So, we start with data, then we build a classification model and once we build the model, we validate the model on test data, so this is an important idea of data science and data analysis. If you use all the data that is given to you and build the model, the model is very likely to do well because it has been all the data we had. We can use predictions as we see more samples or more data that comes in, so we want to have some notion of how well our algorithm is likely to do when we get new data, so to identify that, what we do is we basically split the data into what we call a train and test data and when we build a model, we basically build a model using the train data and then we test this model based on the test data as shown in figure 2. If our model does quite well on the test data, then we can be sure that this model is likely to work in the future.

4.3 Building Logistic Regression Model

First, we will build the logistic regression model for the given data set with dimensions 5000 rows and 14 columns or variables. We can see the columns using the command *print(data.columns)* and we get output as shown in figure 3.

```
In [3]: print(data.columns)
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal Loan', 'Securities Account',
       'CD Account', 'Online', 'CreditCard'],
      dtype='object')
```

Fig. 3. Columns in the given data set

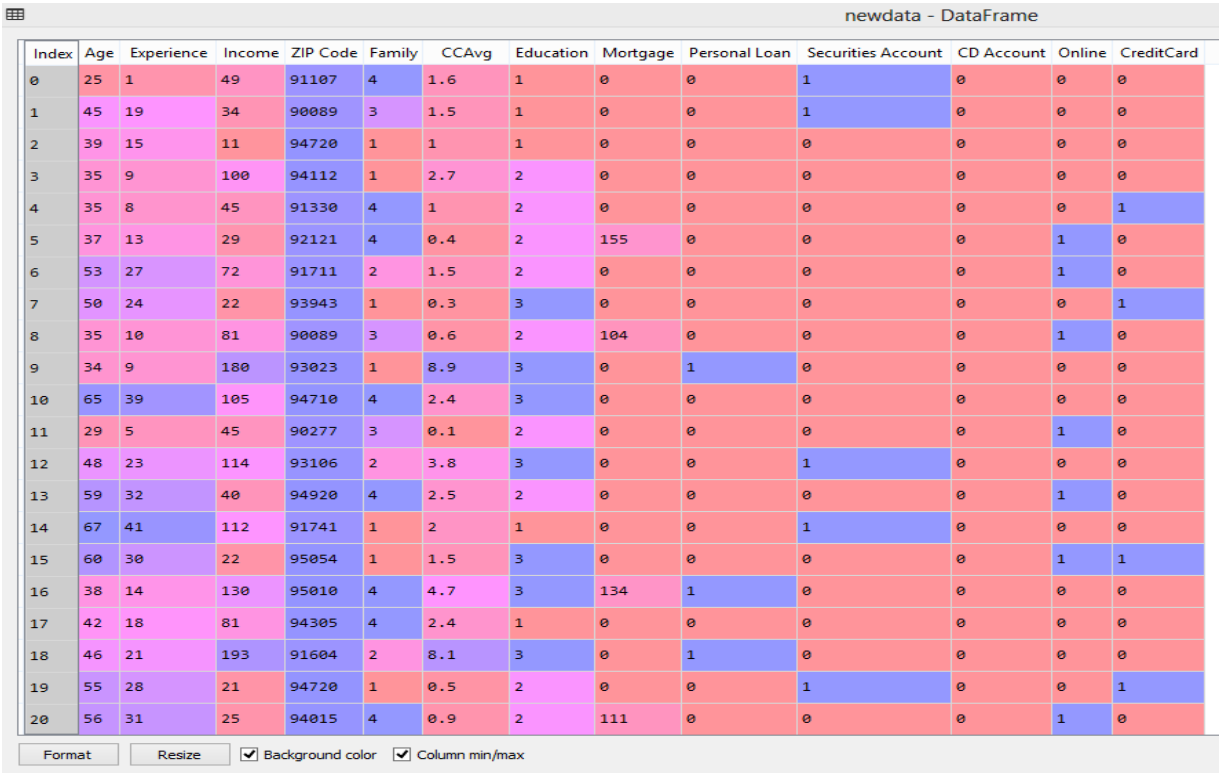
Next step is data processing or removing of all the missing values. A data set may contain categorical and object data types and filling up all the missing values is done differently in each case. A numerical variable should be imputed with mean or median value, whereas a categorical

```
In [4]: print(data.isnull().sum())
ID                   0
Age                  0
Experience           0
Income               0
ZIP Code             0
Family               0
CCAvg                0
Education            0
Mortgage             0
Personal Loan        0
Securities Account   0
CD Account           0
Online               0
CreditCard           0
dtype: int64
```

variable is imputed with the modal value. Missing values in the columns can be known using the command *print(data.isnull().sum())* and we get output as shown in figure 4.

Fig. 4. Number of missing values in the data set

With the following result, we get no missing values and thus we don't need to do data processing. With this we will proceed with building the model. The variable 'ID' does not add any interesting information between a person and loan for future potential loan customers. S we are gonna drop 'ID' column using the command *data2=data.drop('ID',axis=1)* and store the output in new variable, 'data2'. Next, using pandas, we can convert the categorical variable into dummy variables which is known as one hot encoding, using the command *newdata=pd.get_dummies(data2,drop_first=True)* and store the output in variable 'newdata'. We get new data as shown in figure 5.



newdata - DataFrame

Index	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
1	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
2	39	15	11	94720	1	1	1	0	0	0	0	0	0
3	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
4	35	8	45	91330	4	1	2	0	0	0	0	0	1
5	37	13	29	92121	4	0.4	2	155	0	0	0	1	0
6	53	27	72	91711	2	1.5	2	0	0	0	0	1	0
7	50	24	22	93943	1	0.3	3	0	0	0	0	0	1
8	35	10	81	90089	3	0.6	2	104	0	0	0	1	0
9	34	9	180	93023	1	8.9	3	0	1	0	0	0	0
10	65	39	105	94710	4	2.4	3	0	0	0	0	0	0
11	29	5	45	90277	3	0.1	2	0	0	0	0	1	0
12	48	23	114	93106	2	3.8	3	0	0	1	0	0	0
13	59	32	40	94920	4	2.5	2	0	0	0	0	1	0
14	67	41	112	91741	1	2	1	0	0	1	0	0	0
15	60	30	22	95054	1	1.5	3	0	0	0	0	1	1
16	38	14	130	95010	4	4.7	3	134	1	0	0	0	0
17	42	18	81	94305	4	2.4	1	0	0	0	0	0	0
18	46	21	193	91604	2	8.1	3	0	1	0	0	0	0
19	55	28	21	94720	1	0.5	2	0	0	1	0	0	1
20	56	31	25	94015	4	0.9	2	111	0	0	0	1	0

Format | Resize | Background color | Column min/max

Fig. 5. Dummy Variables

Next, we are going to get all the column list and dividing the columns in two types, one having independent variables and one having dependent variables, with dependent variable 'Personal Loan', represented as variable 'y' and with 12 independent variables stored in features list, represented as variable 'x'. Dimensions of x and y are (5000,12) and (5000) respectively.

```
column_list=list(newdata.columns)
features=list(set(column_list)-set(['Personal Loan']))

y=newdata['Personal Loan'].values
x=newdata[features].values
```

Next, we are going to split the data into train set and test set, so that we can build the model on train set and test set to test the model on. For this we must import train_test_split library as: *from sklearn.model_selection import train_test_split* , this is done using the command *train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.3,random_state=0)* , where x represent input values and y represent the output values and test_size is set to 0.3, which

represent the proportion of data set to include in the test split and random_state is set to 0, so that each time we run the model, same data samples is chosen for analysis. Test set have about 30% and train set about 70% of data set as shown in figure 6.

test_x	float64	(1500, 12)	[[1.00000000e+00 5.40000000e+01 4.00000000e-01 ... 0.00000000e+00 0. ...
test_y	int64	(1500,)	[0 0 0 ... 0 0 0]
train_x	float64	(3500, 12)	[[1. 36. 7.8 ... 0. 1. 158.] [2. 59. 3. ... 0 ...
train_y	int64	(3500,)	[0 1 0 ... 0 0 0]

Fig. 6. Train set and Test set

Now we are doing to create a logistic regression classifier instance by importing LogisticRegression library as *from sklearn.linear_model import LogisticRegression* , instance can be created using the command *logistic=LogisticRegression()* and and stored in variable 'logistic'. Now we will fit the model on the train data using .fit function as: *logistic.fit(train_x,train_y)* with input data(train_x) and output data(train_y). Next step is prediction from test data using .predict function as: *prediction = logistic.predict(test_x)* and the output will have only 1 and 0 as values as shown in figure 7, where 0 means 'non potential customer for loan' and 1 means 'potential customer for loan'.

```
In [6]: prediction = logistic.predict(test_x)

In [7]: print(prediction)
[0 0 0 ... 0 0 0]
```

Fig. 7. Logistic Regression Prediction

Till now we have made the model using train set and tested it on a new data set, that is, test_x and got the predictions.

4.4 Building k-Nearest Neighbor Model

Now we will build a kNN classifier model, to classify the records into anyone of the categories of personal loan. To build the kNN model, we are going to import KNeighborClassifier from sklearn.neighbor as : *from sklearn.neighbors import KNeighborsClassifier*

Next we will create an instance for kNN classifier using the function KNeighborClassifier and inside the function we will specify n_neighbors as 5, that is the value of k, so that it will consider 5 neighbors when classifying the data into potential customer and non-potential customer for

loan. The output is saved into a variable 'kNN_classifier' as: *kNN_classifier = KNeighborsClassifier(n_neighbors=5)*

So, if it considers 5 neighbors then it will take the majority classes from the 5 neighbors and then it will classify the new data based on the majority voting matter and hence, we get our model as kNN_classifier. Now we will fit the model on the data frame using .fit function with input data(train_x) and output data(train_y). Next step is prediction from the test data using .fit function, the output will have 0 and 1 as values, where 0 means 'non potential customer for loan' and 1 means 'potential customer for loan'. The output is saved in the variable 'prediction2' as: *prediction2 = kNN_classifier.predict(test_x)* as shown in figure 8.

```
In [25]: from sklearn.neighbors import KNeighborsClassifier

In [26]: kNN_classifier = KNeighborsClassifier(n_neighbors=5)

In [27]: kNN_classifier.fit(train_x,train_y)
Out[27]:
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')

In [28]: prediction2 = kNN_classifier.predict(test_x)

In [29]: print(prediction2)
[0 0 0 ... 0 0 0]
```

Fig. 8. kNN Classifier

5 Experimental Results

This section is about evaluation of model. We are going to evaluate the Logistic Regression model and the k-Nearest Neighbor model based on Confusion Matrix, Accuracy Score and the number of misclassified values from predictions.

- Confusion Matrix : It is a matrix used to evaluate a classification model. The output of confusion matrix gives us the number of correct predictions and the number of incorrect predictions. The input of the confusion matrix are the actual values, that is, test_y and the predictions made. The diagonal values give you the total number of correctly classified samples and the off diagonal gives you the total number of wrongly classified samples. In column, we have the predictions and in rows, we have the actual classes. This is done using the command *confusion_matrix=confusion_matrix(test_y,prediction)* as shown in figure 9 and 10.

```
In [11]: confusion_matrix=confusion_matrix(test_y,prediction)

In [12]: print('Confusion Matrix for Logistic Regression\n',confusion_matrix)
Confusion Matrix for Logistic Regression
 [[1329   43]
 [  78   50]]
```

Fig. 9. Confusion Matrix of Logistic Regression

```
In [30]: confusion_matrix=confusion_matrix(test_y,prediction)

In [31]: print('Confusion Matrix for k-Nearest Neighbor\n',confusion_matrix)
Confusion Matrix for k-Nearest Neighbor
 [[1331   41]
 [  91   37]]
```

Fig. 10. Confusion Matrix of k-Nearest Neighbor

- Accuracy Score : Using a measure called accuracy, we will be able to get accuracy score of the model and it tells us how accurately our model is working. The input of the accuracy is the actual class, that is, test_y and the predictions made in the model. Informally, accuracy is the fraction of predictions our model got right. This can be done using the command *accuracy_score=accuracy_score(test_y,prediction)* as shown in figure 11 and 12.

```
In [7]: accuracy_score=accuracy_score(test_y,prediction)

In [8]: print('Accuracy Score for Logistic Regression\n',accuracy_score)
Accuracy Score for Logistic Regression
 0.9193333333333333
```

Fig. 11. Accuracy score of Logistic Regression

```
In [4]: accuracy_score=accuracy_score(test_y,prediction)

In [5]: print('Accuracy Score for for k-Nearest Neighbor\n',accuracy_score)
Accuracy Score for for k-Nearest Neighbor
 0.912
```

Fig. 12. Accuracy score of k-Nearest Neighbor

- Misclassified values : In this we will find the number of misclassifid values from the predictions made. This means it will the number of wrongly predicted samples as the output. This can be done by giving a simple condition, test_y! = predictions. The command for this is *print('Misclassified samples : %d'% test_y! = prediction.sum())* as shown in figure 13 and 14.

```
In [12]: print('Misclassified samples of Logistic Regression model : %d' % (test_y !=
prediction).sum())
Misclassified samples of Logistic Regression model : 121
```

Fig. 13. Misclassified samples of Logistic Regression

```
In [5]: print('Misclassified samples of k-Nearest Neighbor model : %d' % (test_y != prediction).sum())
Misclassified samples of k-Nearest Neighbor model : 132
```

Fig. 14. Misclassified samples of k-Nearest Neighbor

6 Conclusion

This study introduces two machine learning techniques, one is the Logistic Regression and other is the k-Nearest Neighbor and with respect to the each method, a model is built for campaign for bank personal loan and this provides a guideline for similar comparison studies. Evaluation is done based on confusion matrix, accuracy, and number of misclassified values. Based on results, we got, we can see that :

- In confusion matrix, logistic regression's correctly predicted samples are equal to 1379(1329+50) and k-nearest neighbor's correctly predicted samples is equal to 1368(1331+37), that is, logistic regression gives more correctly predicted samples than k-nearest neighbor.
- In accuracy part, with the results we can see that accuracy of logistic regression is equal to 0.919 and that of k-nearest neighbor is equal to 0.912, that means logistic regression gives us the more accurate model.

- In case of misclassified values, the number of misclassified values given by logistic regression is 121, whereas for k-nearest neighbor it gets increased to 132.

Therefore, with all these results, we can conclude that, for a binary classification problem of bank personal loan, Logistic Regression gives us the more accurate and efficient model.

References

[1] Nigel Williams, Sebastian Zander, Grenville Armitage: A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification. www.researchgate.net. (2006)

[2] Bichler, Martin and Kiss, Christine: A Comparison of Logistic Regression, k-Nearest Neighbor, and Decision Tree Induction for Campaign Management. AMCIS 2004 Proceedings. (2004)

[3] Roger D. Peng and Elizabeth Matsui: The Art of Data Science. Lean Publishing. (2015)

[4] Ling Li, Hsuan-Tien Lin: Ordinal Regression by Extended Binary Classification**.** Learning Systems Group,California Institute of Technology.

[5] Mohamed Aly: Survey on Multiclass Classification Methods. malaa@caltech.edu. (2005)

[6] Stephen Marsland: Machine Learning an Algorithmic Perspective: CRC Press Taylor & Francis Group. (2015)

[7] Ralf Bender,Ulrich Grouven: Using Binary Logistic Regression Models for Ordinal Data with Non-proportional Odds. Elsevier Science Inc. (1998)

[8] Dávid Natingga: Data Science Algorithms in a Week: Birmingham, Packt Publishing Ltd. (2017)

[9] Chao-Ying, Joanne Peng, Kuk Lida Lee, Gary M. Ingersoll: An Introduction to Logistic Regression Analysis and Reporting. The Journal of Educational Research. (2005)

[10] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer: KNN Model-Based Approach in Classification. European Commission project ICONS. project no. IST-2001-32429.

[11] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer: An KNN Model-Based and its applications in Text Categorization. European Commission project ICONS. project no. IST-2001-32429.

[12] https://www.kaggle.com/itsmesunil/bank-loan-modelling/version/1#

[13] M.Palazzo and R. Evans: Logistic Regression Analysis of Fixed Patient factors for Postoperative Sickness : A Model for Risk Assessment. British Journal of Anaesthesia; 70: 135-140. (1993)

[14] F.Y. Hseih, Daniel A. Bloch and Michael D. Larsen. A Simple Method of sample size calculation for Linear and Logistic Regression. John Wiley & Sons, Ltd. (1998)

[15] Shichao Zhang, Senior Member, IEEE, Xuelong Li, Fellow, IEEE, Ming Zong, Xiaofeng Zhu, and Ruili Wang: Effificient kNN Classifification With Different Numbers of Nearest Neighbors. IEEE Transactions on Neural Networks and Learning Systems; Vol. 29. No. 5. (2018)

[16] Gilpin, A.E., and Ohlsen, R.A.: Predicting 1-Year Outcome following Acute Myocardial Infarction: Physicians versus Computers, Computers and Biomedical Research, pp 46-63. (1990)

[17] Michie, D., Spiegelhalter, D.J., and Taylor, C.C.: Machine Learning, Neural and Statistical Classification Ellis Horwood. New York. (1994)