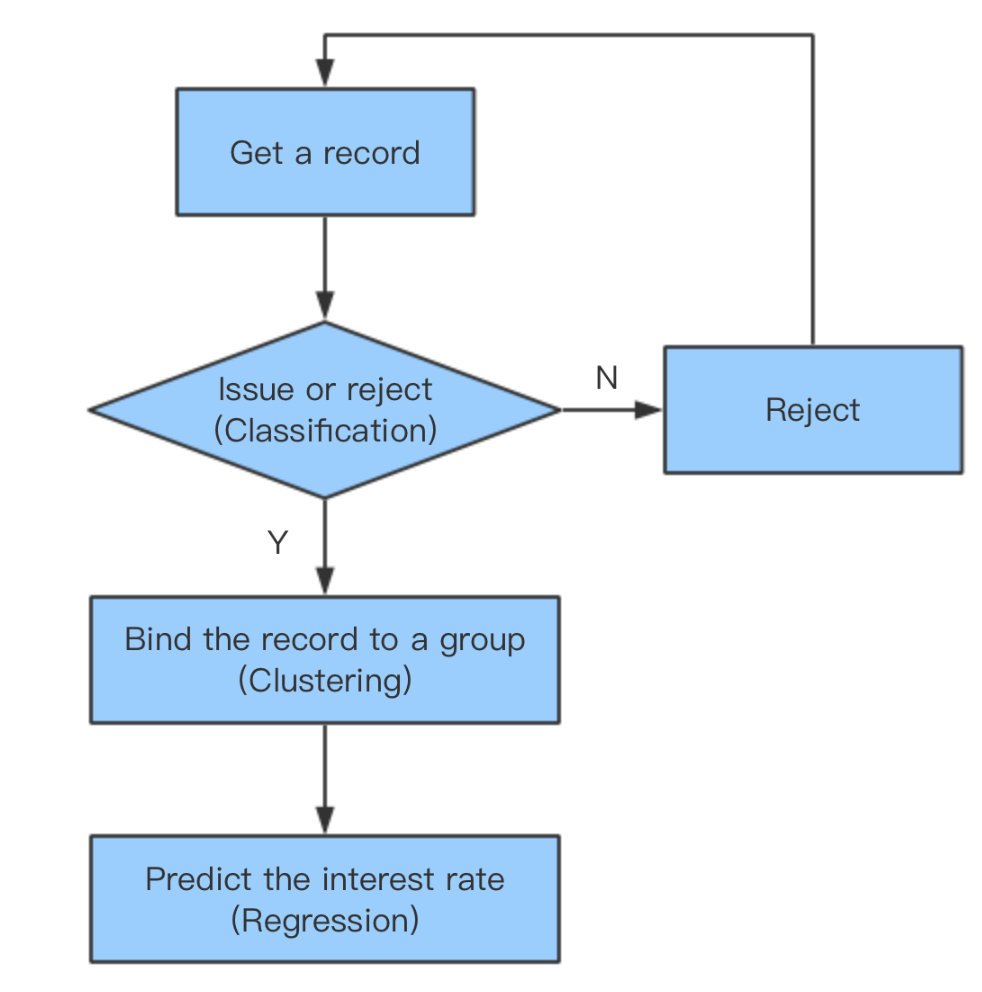
**Loan Club Data Analysis**

Jiali Cheng, Sicheng Zhang

1. Introduction

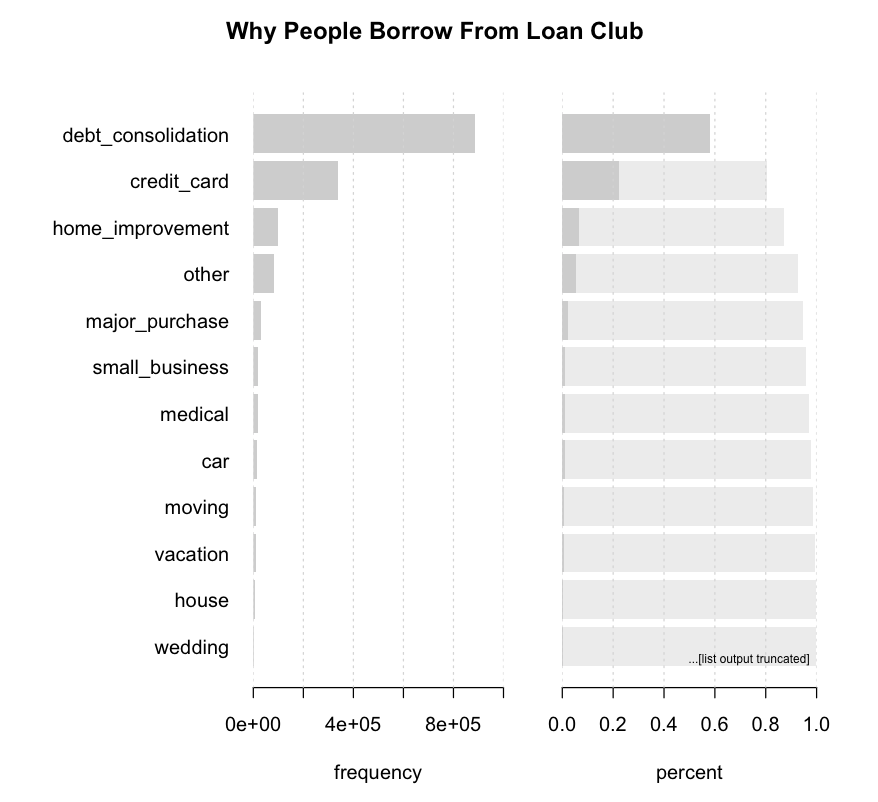
In this project, we are dealing with Loan Club data. We will build a machine learning model to do the following job. Given a loan request, or a record, we should first decide whether or not issue it. If issue, we should decide the interest rate for this request. The following graph shows the process.



1. Exploratory Data Analysis

After getting the data, we first look at its summary, containing object type and few values of each column.

We had a look at the loan title and purpose, which indicates why people borrow a loan. And also print the word cloud.

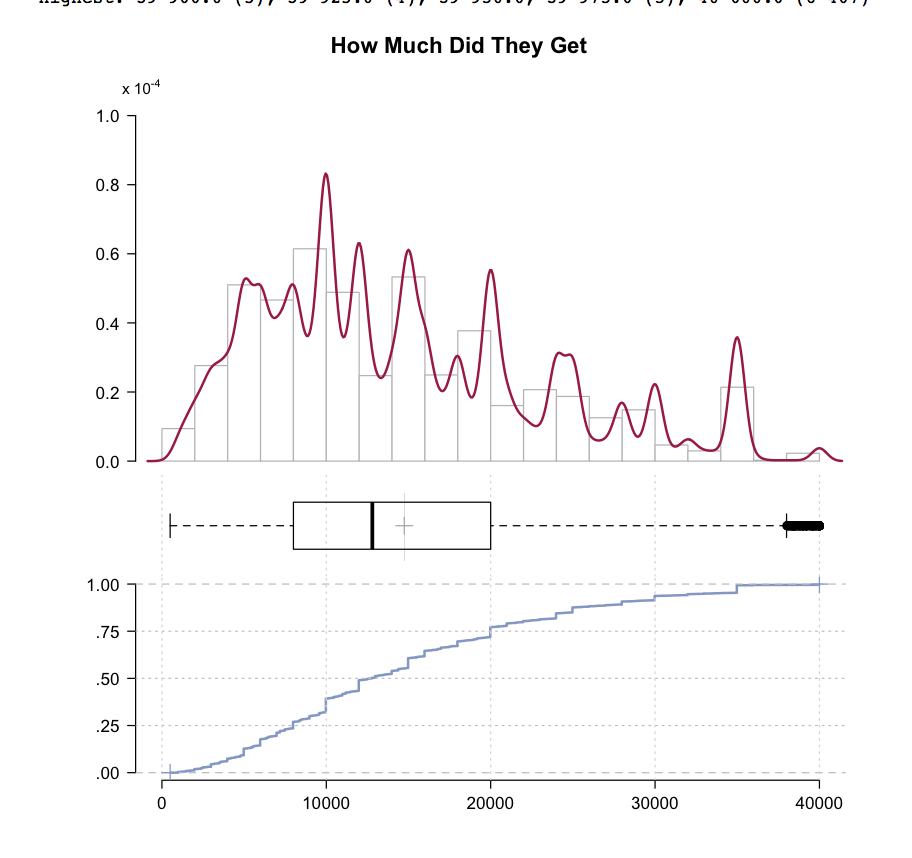




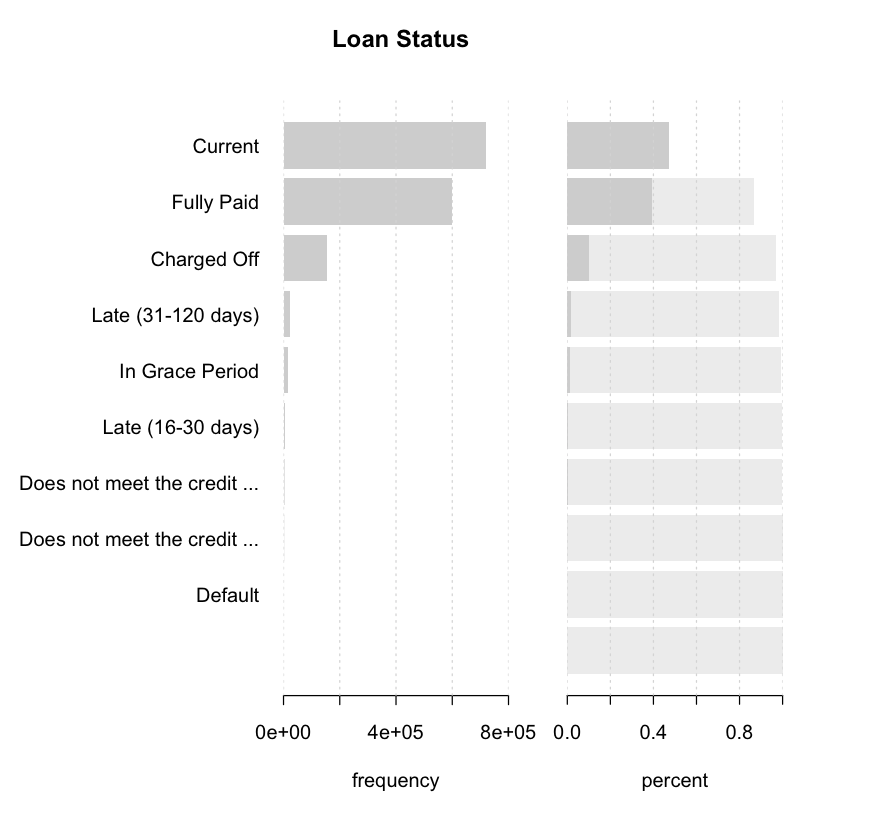
and what title people used to borrow the loan.



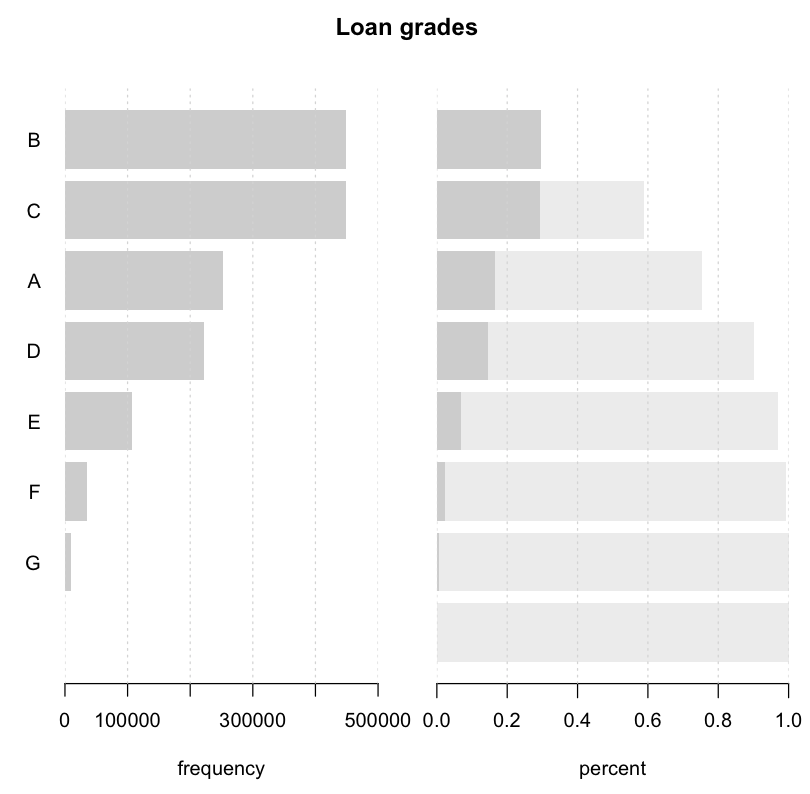
and how much do they get.



and the loan status.



and the loan grade.



We grouped the data in terms of state and calculate the total issued loan amount and number of issued loans within each state.

1. Data Cleaning and Preprocessing
2. Cleaning

In data cleaning part we do two major jobs, dealing with missing values and cleaning outliers. We fill missing values with respect to the distribution of data. And we do data cleaning based on two information, the physical realistic and statistical characteristics. If someone’s annual income is 1,000,000,000, it is probably an outlier.

(2) Selecting Features

Given a borrowing loan, deciding to issue or not only uses the features containing in the Rejected Data, since those loan requests were rejected based on only the features in Rejected Data records. And whether a request is going to be rejected has nothing to do with the requesting date. Therefore, for the classification part, we select the “loan amount”, “loan purpose”, “Debt-to-income Ratio”, “Addressed State” plus a column target of “0” and “1” with “0” representing rejected and “1” representing issued.

For the clustering part, however, data set is different, since we are looking inside the issued loan request and want to decide the amount. So the data set is the the issued records. We also did some necessary preprocessing on the data which will be talked about in the later part.

(3) Encoding categorical data

There are categorical data in the training set, such as “addr\_state” indicating the state and “purpose” indicating the purpose of borrowing loan.

In order to apply classification algorithm on the data, we need to do some proper transformation on the categorical features. We tried *Label Encoding* and *Binary Encoding* and compared their performance under same classification algorithm.

Our targets of encoding are features “addr\_state” with 52 possible values and “purpose” with 15 possible values.

Label Encoding assigns a number for each possible value and does not change the dimension of the training set. While Binary Encoding extends one column to the number of its possible values consists of “0”s and “1”s which is a sparse matrix. So we use PCA to select important components for the benefit of saving computing resources and time.

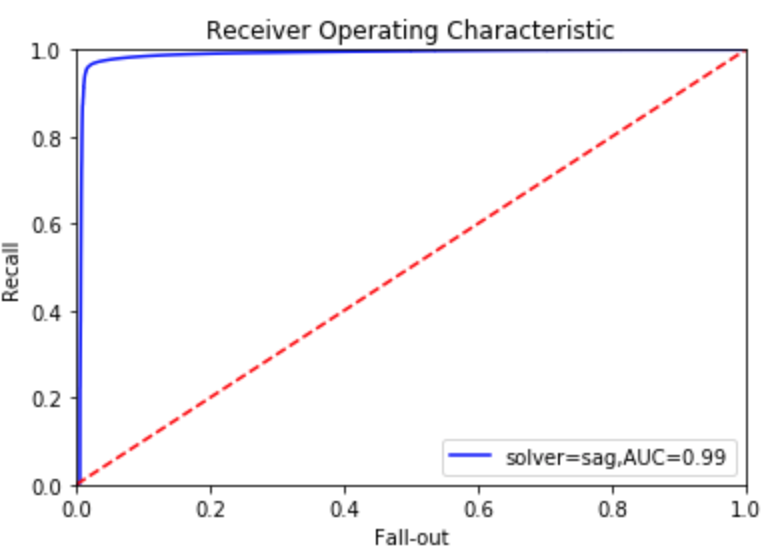
1. Classification

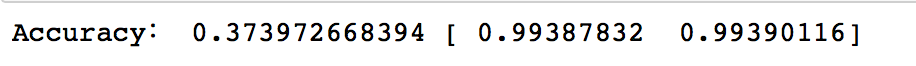
In this section, we deal with the problem of judging whether or not giving a person loan.

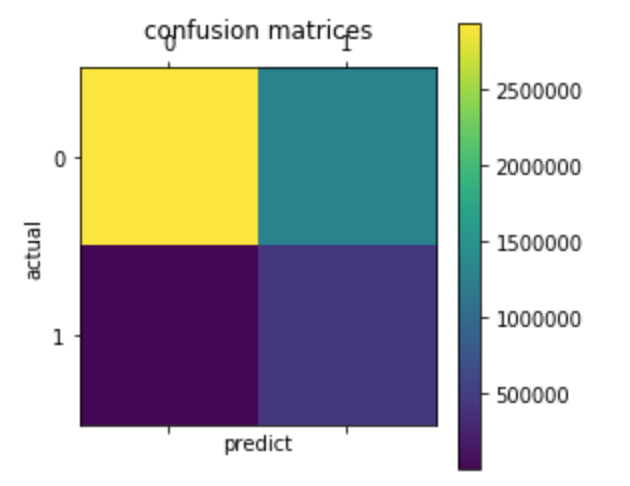
As is shown in the rejected records, there are only features of date, loan amount, purpose, state, risk score and loan title. This indicates that we can simply figure out whether issuing a specific loan by only these pieces of information, which extremely decreases the data set we have to process.

4.1 Logistic regression

The results are as below. We can see that the accuracy is very high.

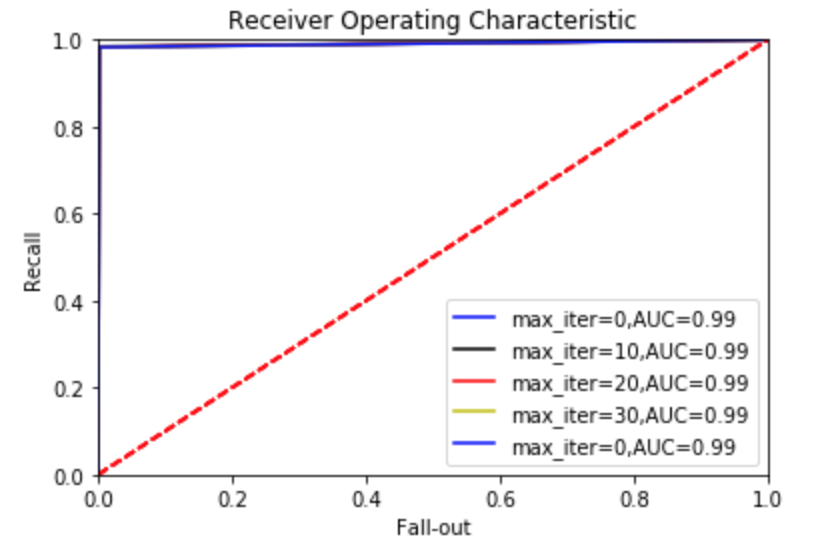




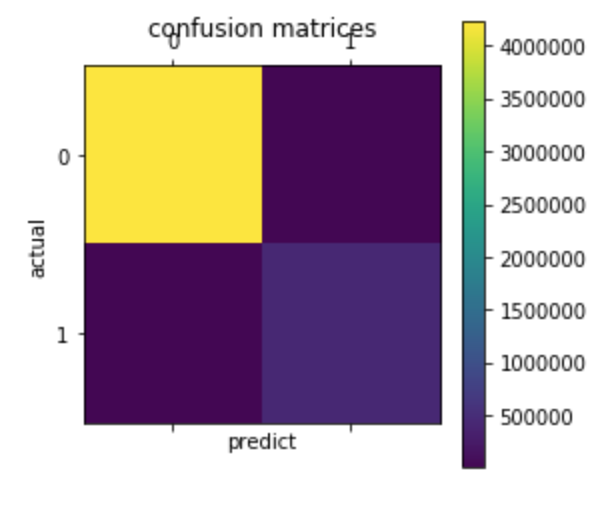


4.2 Random forest

The result is as follow.



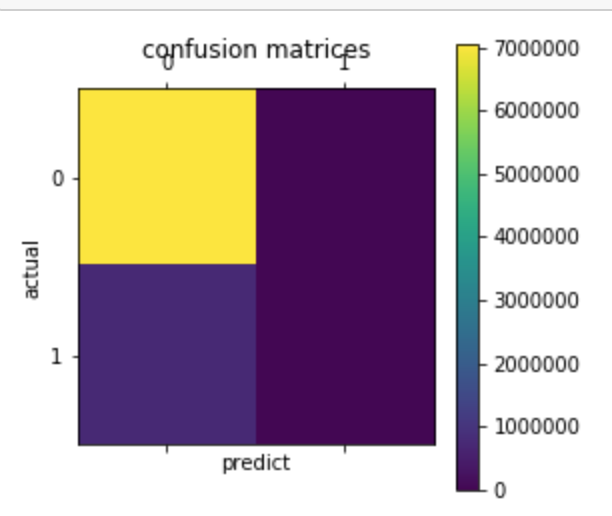




As we can see from the figures, the accuracy is very high. And TPR, TFR are very high.

4.3 Neural network

Neural network is giving a terrible result. It may be because the network is too shallow.



1. Clustering and Prediction

Now after deciding to give a specific loan, we have to decide how much amount we should give. There is no strict rule we can learn from the data set, which is an unsupervised learning problem. We apply 3 different schemes of clustering which are manually clustering in terms of specific feature(s), clustering with ML algorithm and no clustering and then predict the interest rate for each cluster.

For each cluster, we apply regression models of linear regression, random forest, neural networks and KNN to predict the interest rate of the loan. And we use metric of RMS, MAE and MAPE to evaluate the models, whose values are shown below. Let’s start with manually clustering.

* 1. Manually clustering

By observing the data set carefully, we tried clustering the loan records into 2 clusters in terms of the loan term.

* 1. Algorithmic clustering

The data set is dissimilar, which means it contains numeric values and factors. And we have to use algorithms that can handle input of mixed type.

We use hierarchical clustering (Agglomerative Nesting) which can handle input of mixed type. We tried different metrics for clustering, including Gower’s dissimilarity coefficient and Ward’s algorithm. Gower's coefficient (1971), expressed as a dissimilarity, implies that a particular standardization will be applied to each variable, and the “distance” between two units is the sum of all the variable-specific distances, see the details section.

There are several methods we could use. We list them and compare them.

Daisy: Compute all the pairwise dissimilarities (distances) between observations in the data set.

Hierarchical methods:

Agnes: Computes agglomerative hierarchical clustering of the dataset.

Hclust: Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.

Partitioning methods:

Pam: Partitioning (clustering) of the data into k clusters “around medoids”, a more robust version of K-means. All input values must be numeric.

Compared to the k-means approach in kmeans, the function pam has the following features: (a) it also accepts a dissimilarity matrix; (b) it is more robust because it minimizes a sum of dissimilarities instead of a sum of squared euclidean distances; (c) it provides a novel graphical display, the silhouette plot (see plot.partition) (d) it allows to select the number of clusters using mean(silhouette(pr)[, "sil\_width"]) on the result pr <- pam(..), or directly its component pr$silinfo$avg.width, see also pam.object.

Clara: a list representing a clustering of the data into k clusters. Internally, this is achieved by considering sub-datasets of fixed size (sampsize) such that the time and storage requirements become linear in n rather than quadratic.

Each sub-dataset is partitioned into k clusters using the same algorithm as in pam.  
Once k representative objects have been selected from the sub-dataset, each observation of the entire dataset is assigned to the nearest medoid.

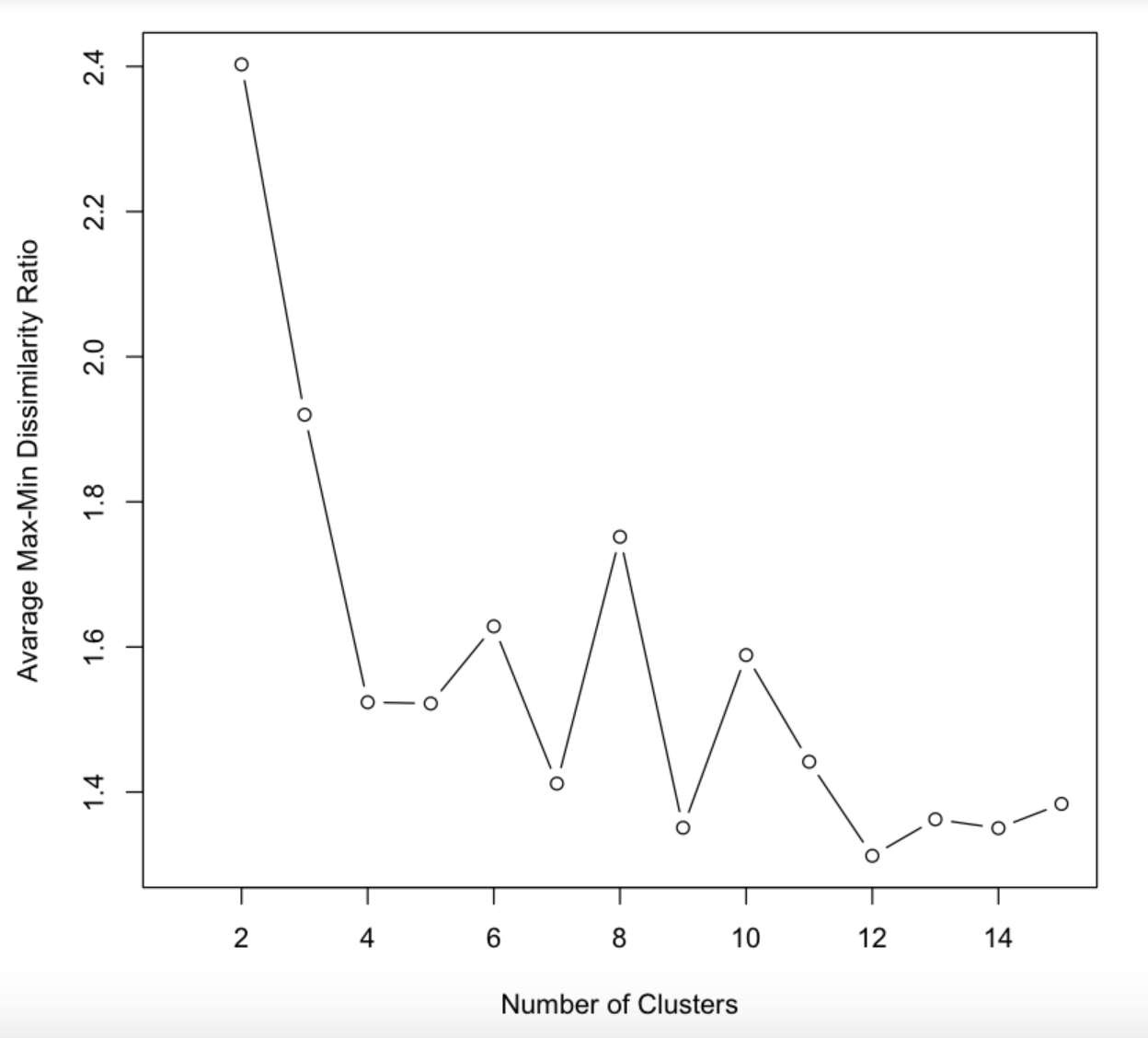
The metric of Gower distance used for each data type are shown below:

Quantitative: range-normalized Manhattan distance

Ordinal: variable is first ranked, then Manhattan distance is used with a special adjustment for ties

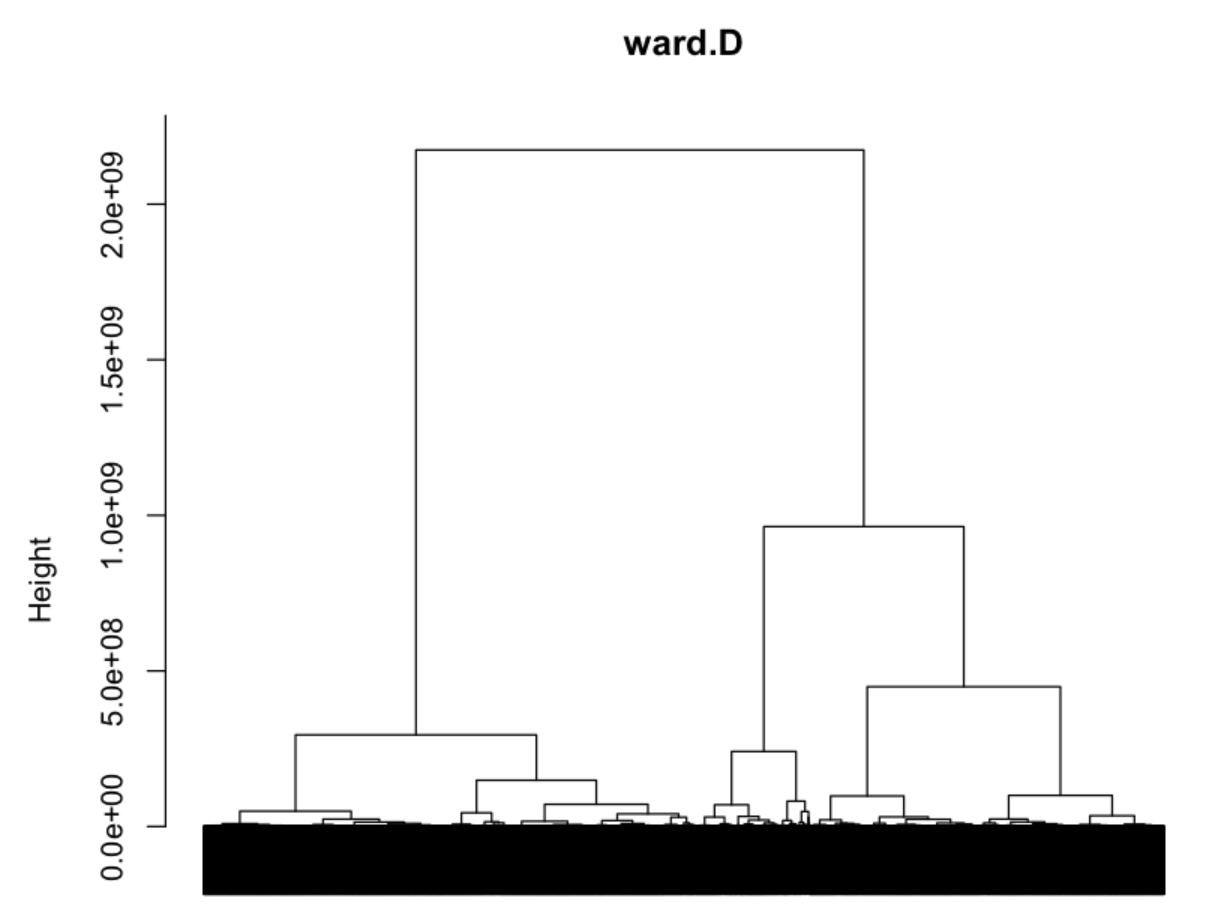
Nominal: variables of k categories are first converted into k binary columns

Then we plot the bend graph to see how many clusters is the optimal choice.



From the graph we can see that from 2-4 clusters, the error drops heavily. While the number grows to 5, the error changes little. Though there are other numbers of clusters that have attained higher performance, we choose 5 clusters for the sake of computation time. And 5 cluster is good enough for this problem. Also some clusters such as cluster 7, has too less data in it. Instead of using pam, we use daisy to compute the dissimilarity of the variable and then throw it into clara algorithm, because clara is much more efficient than pam.

We also tried hclust below. But clara is better.



* 1. No cluster

The third scheme is simply use the raw data without clustering.

1. Prediction

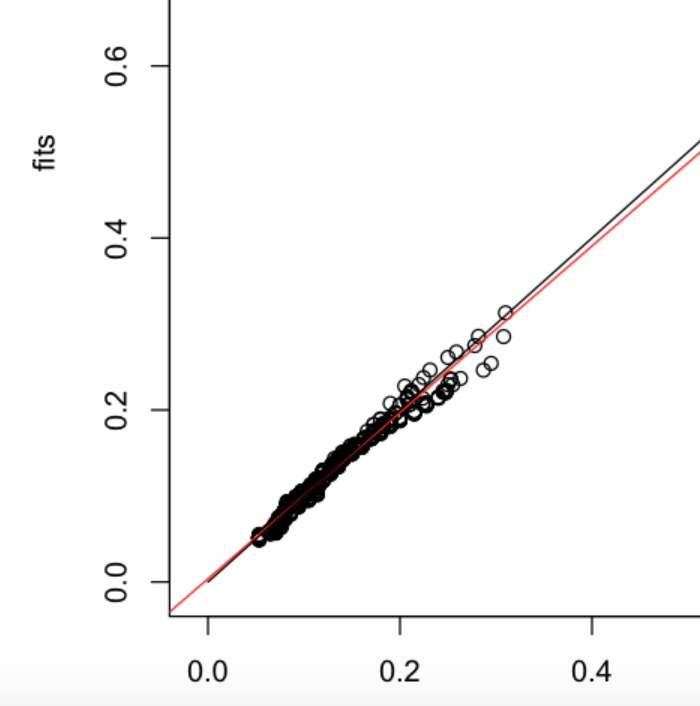
For each scheme of clustering, we do feature selection and use linear regression, random forest, neural network and KNN to predict the interest rate we should give.

6.1 Manually cluster

The features selected are annual\_inc, dti, last\_pymnt\_amnt, mths\_since\_rcnt\_il, all\_util, total\_rev\_hi\_lim, inq\_last\_12m, percent\_bc\_gt\_75, term, grade, sub\_grade, verification\_status and revol\_util. Then we cleaned outliers of

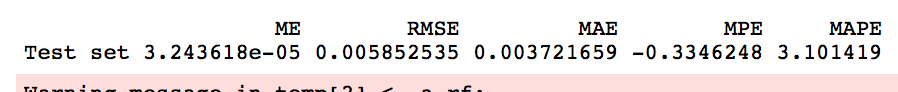
Cluster 1:

For linear regression, the results are as follow.

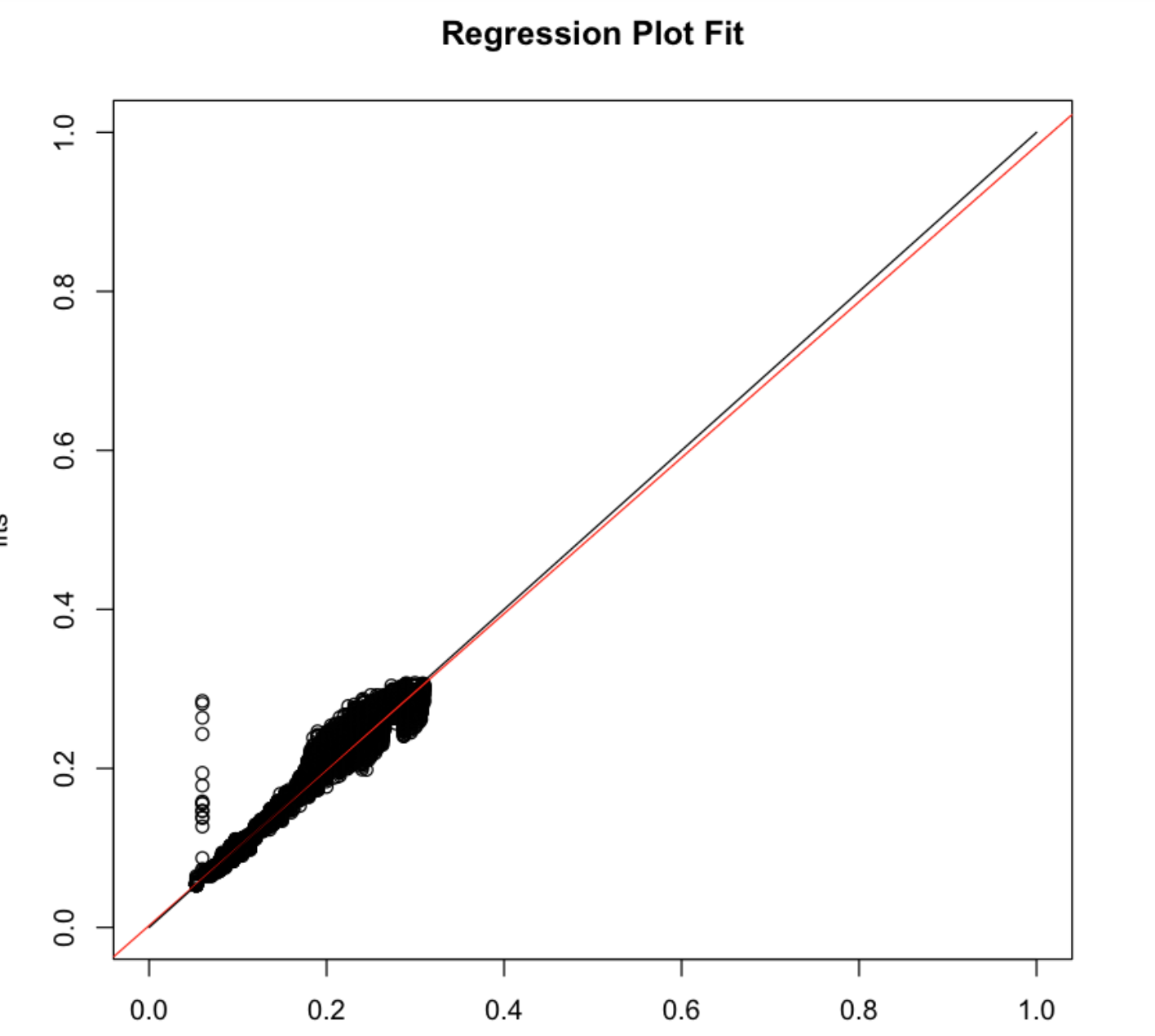


We can see from the results that all the features we put into the algorithm have high significance towards the target. And residuals approximately follow a Normal distribution. And the predicted values are located closely to the real values.

Then we compute metrics below.

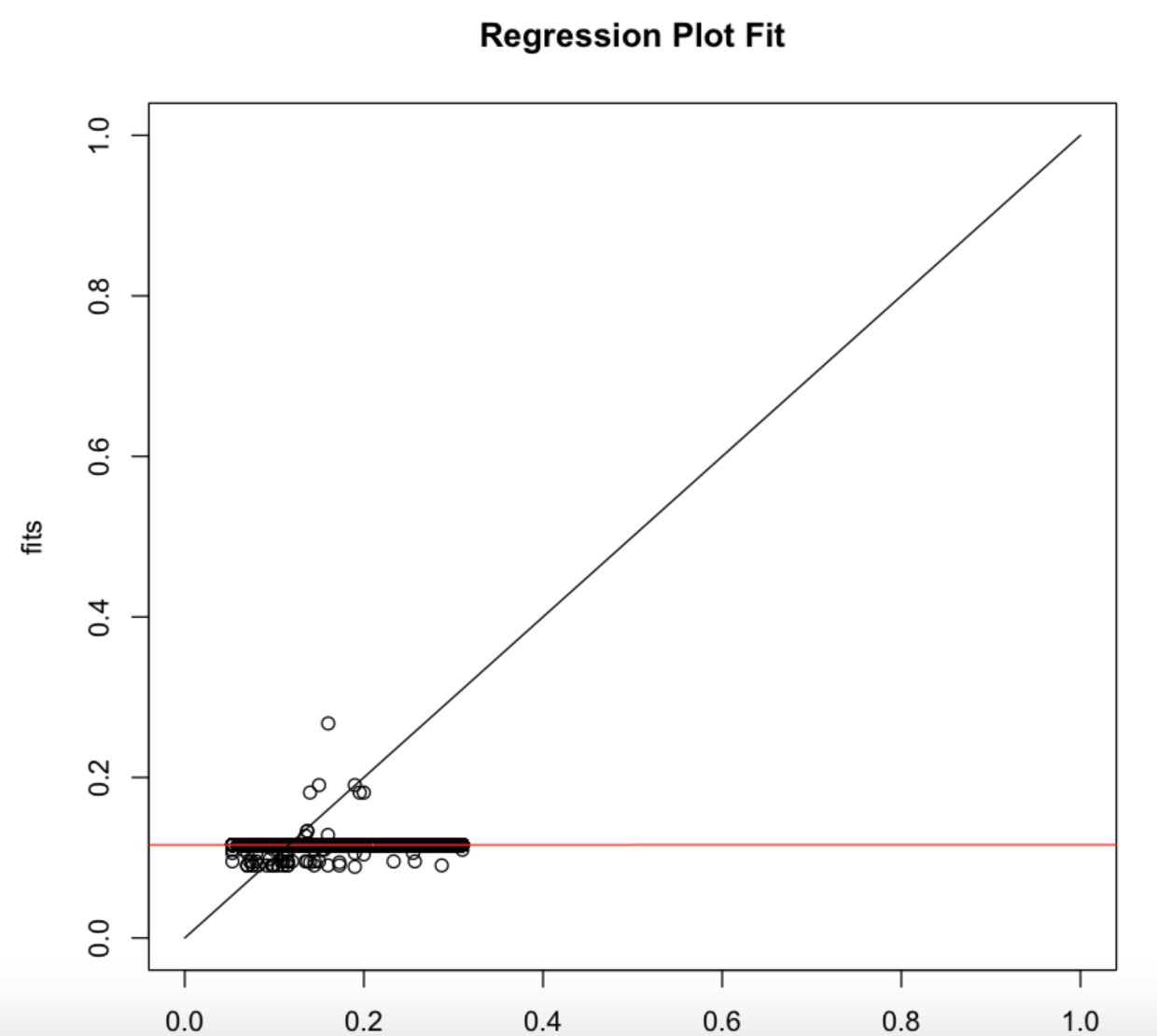


For random forest, the results are as follow.



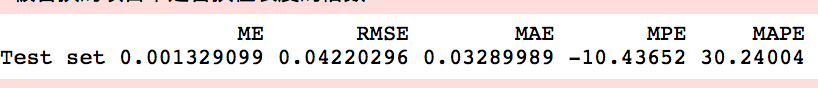
The predicted values are located closely around the real values, except for some outliers.

For neural network, the results are as follow.



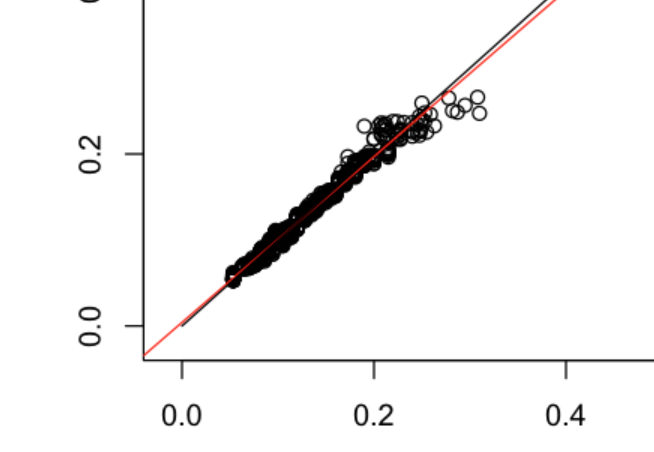
As we can see, neural network is giving the worst result.

For KNN, we use the 10 nearest neighbors to predict. The results are as follow.



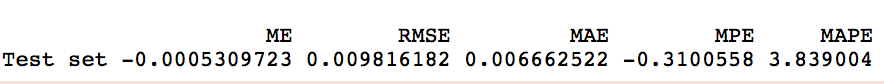
Cluster 2:

For linear regression, the results are as follow.

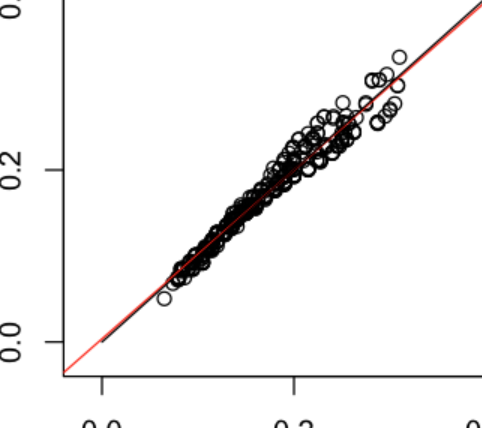


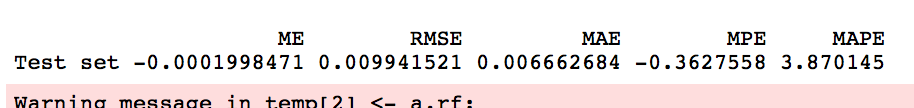
We can see from the results that all the features we put into the algorithm have high significance towards the target. And residuals approximately follow a Normal distribution. And the predicted values are located closely to the real values.

Then we compute metrics below.



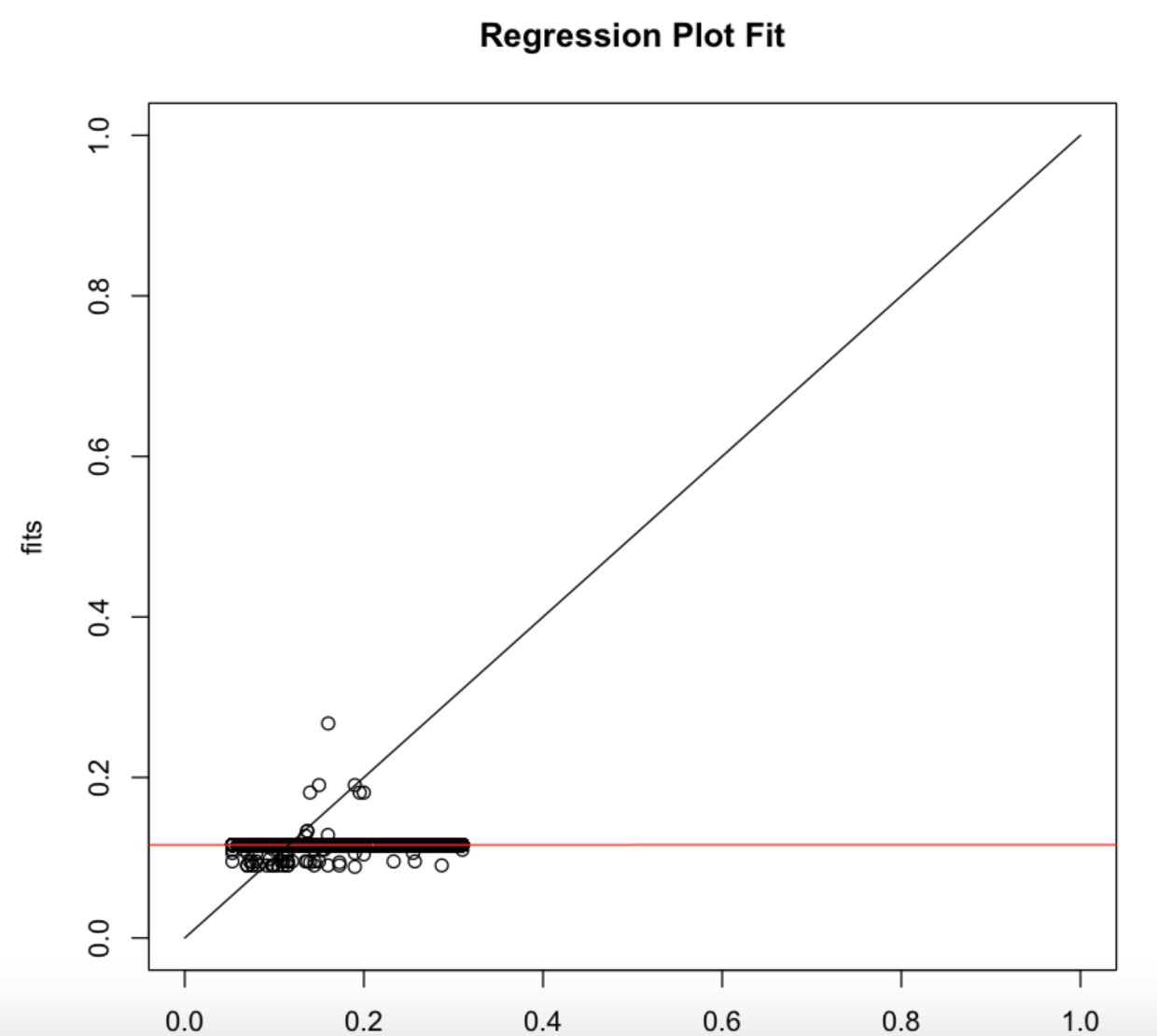
For random forest, the results are as follow.





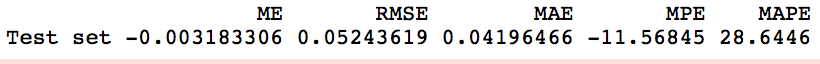
The predicted values are located closely around the real values, except for some outliers.

For neural network, the results are as follow.



As we can see, neural network is giving the worst result.

For KNN, we use the 10 nearest neighbors to predict. The results are as follow.

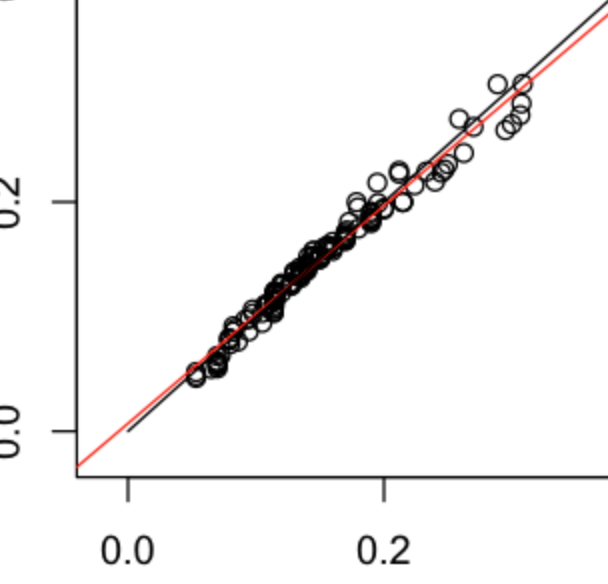


* 1. Algorithmic cluster

The features selected are annual\_inc, dti, last\_pymnt\_amnt, mths\_since\_rcnt\_il, all\_util, total\_rev\_hi\_lim, inq\_last\_12m, percent\_bc\_gt\_75, term, grade, sub\_grade, verification\_status and revol\_util. Then we cleaned outliers of

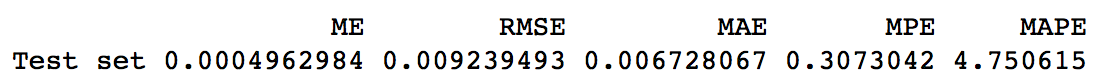
For saving the space, we are just showing one cluster here.

For linear regression, the results are as follow.

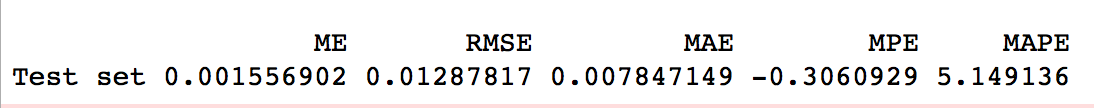


We can see from the results that all the features we put into the algorithm have high significance towards the target. And residuals approximately follow a Normal distribution. And the predicted values are located closely to the real values.

Then we compute metrics below.

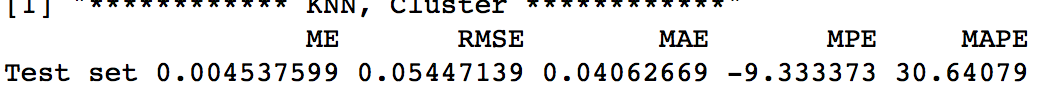


For random forest, the results are as follow.



The predicted values are located closely around the real values, except for some outliers.

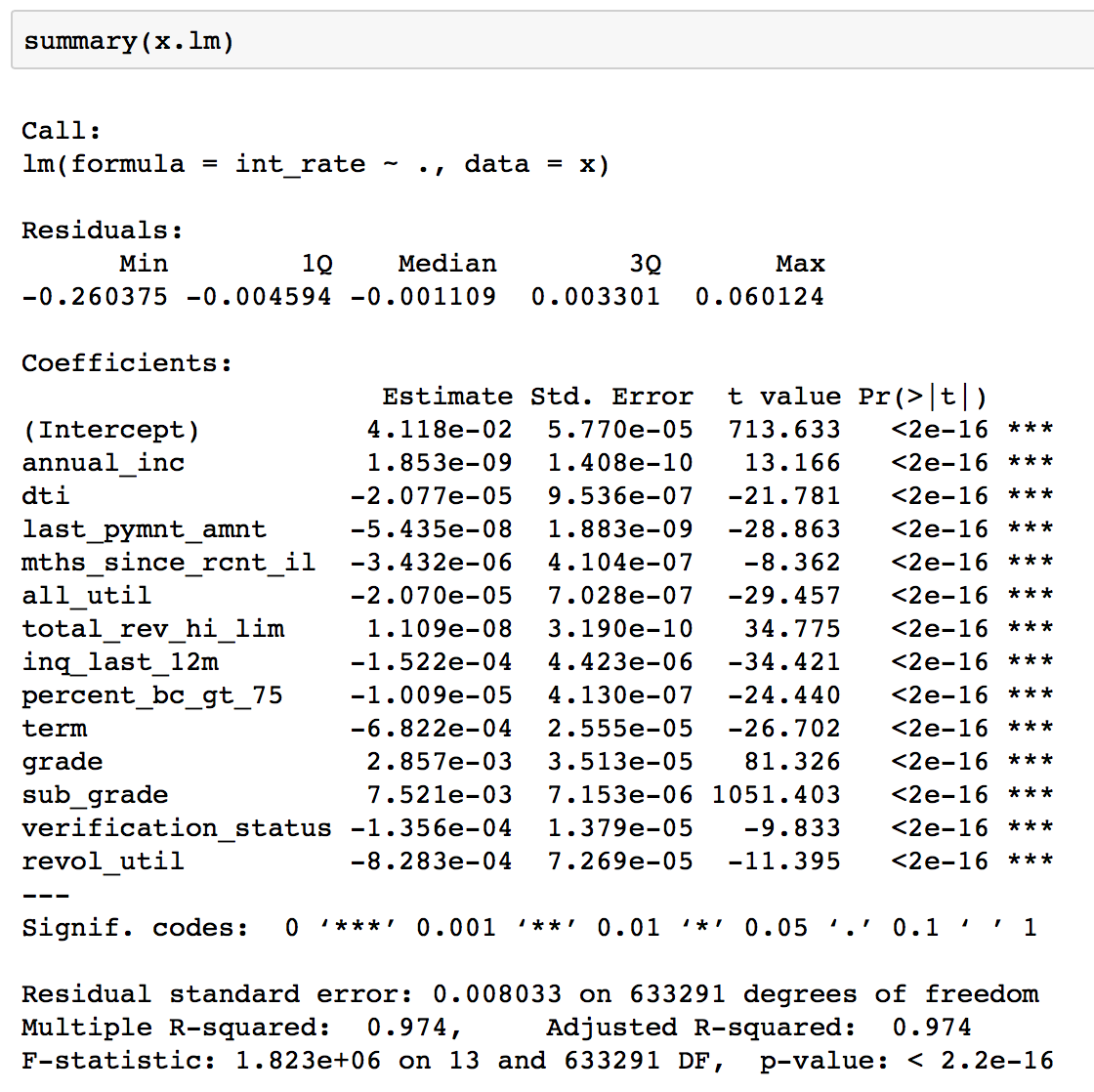
For KNN, we use the 10 nearest neighbors to predict. The results are as follow.

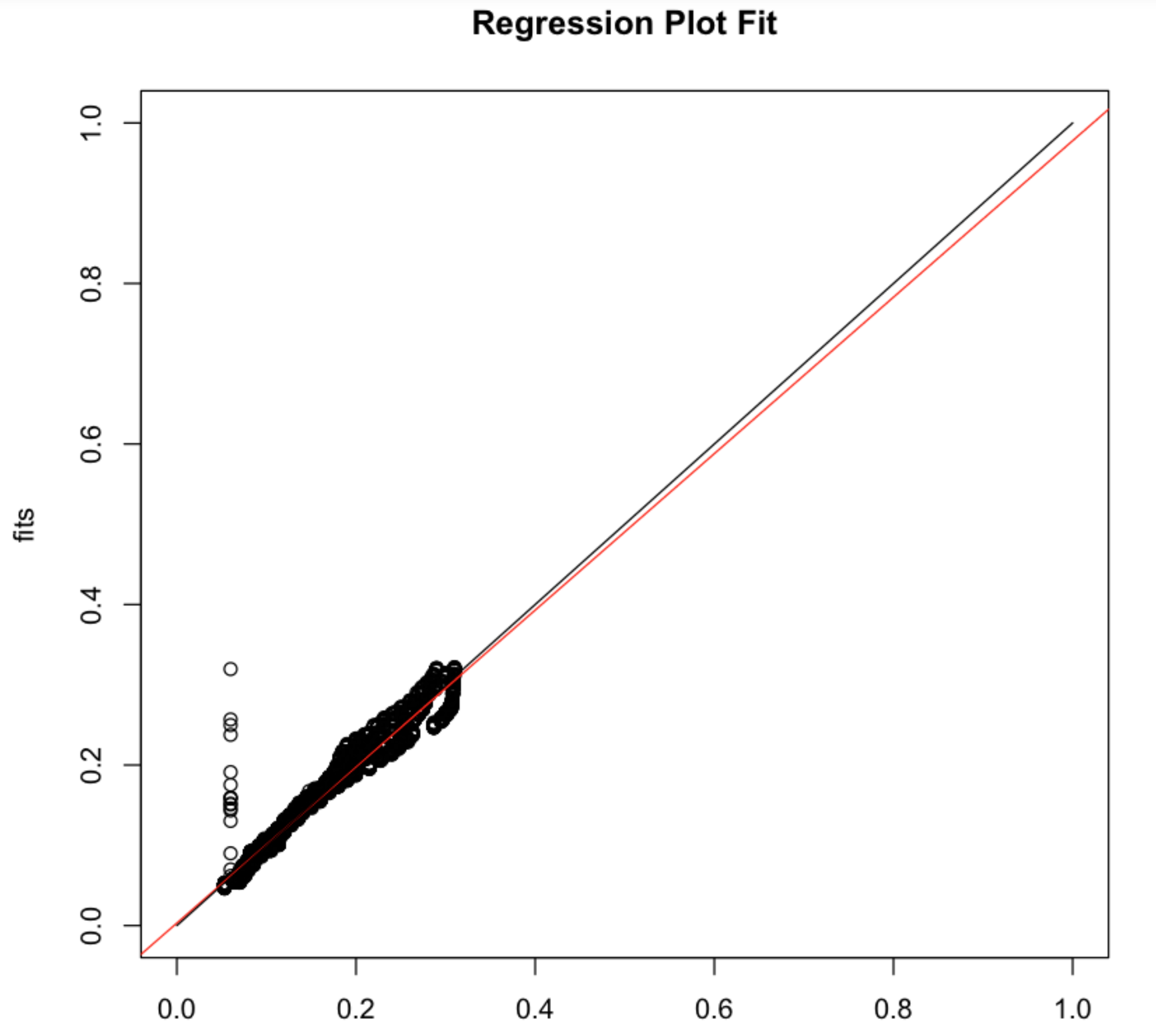


* 1. No cluster

The features selected are annual\_inc, dti, last\_pymnt\_amnt, mths\_since\_rcnt\_il, all\_util, total\_rev\_hi\_lim, inq\_last\_12m, percent\_bc\_gt\_75, term, grade, sub\_grade, verification\_status and revol\_util. Then we cleaned outliers of

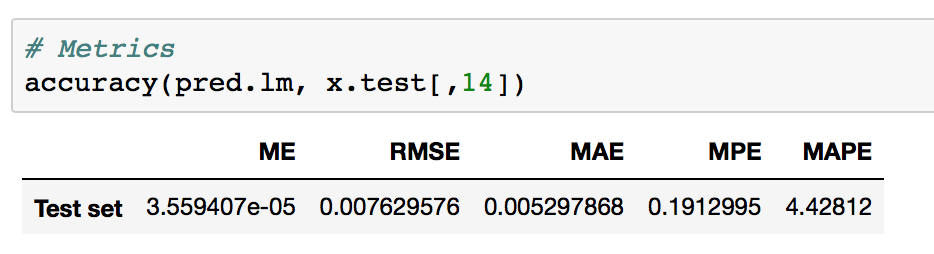
For linear regression, the results are as follow.



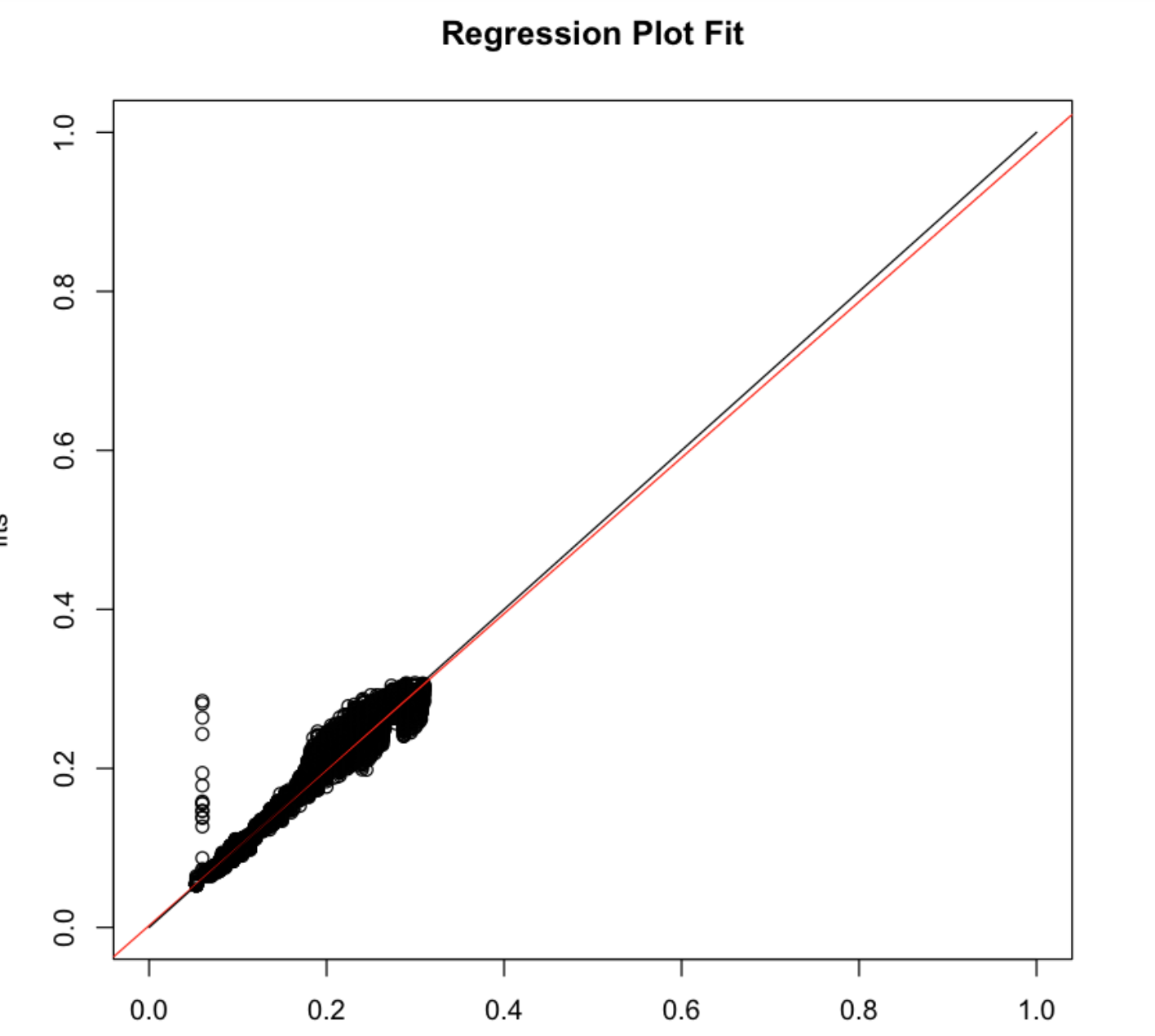


We can see from the results that all the features we put into the algorithm have high significance towards the target. And residuals approximately follow a Normal distribution. And the predicted values are located closely to the real values.

Then we compute metrics below.

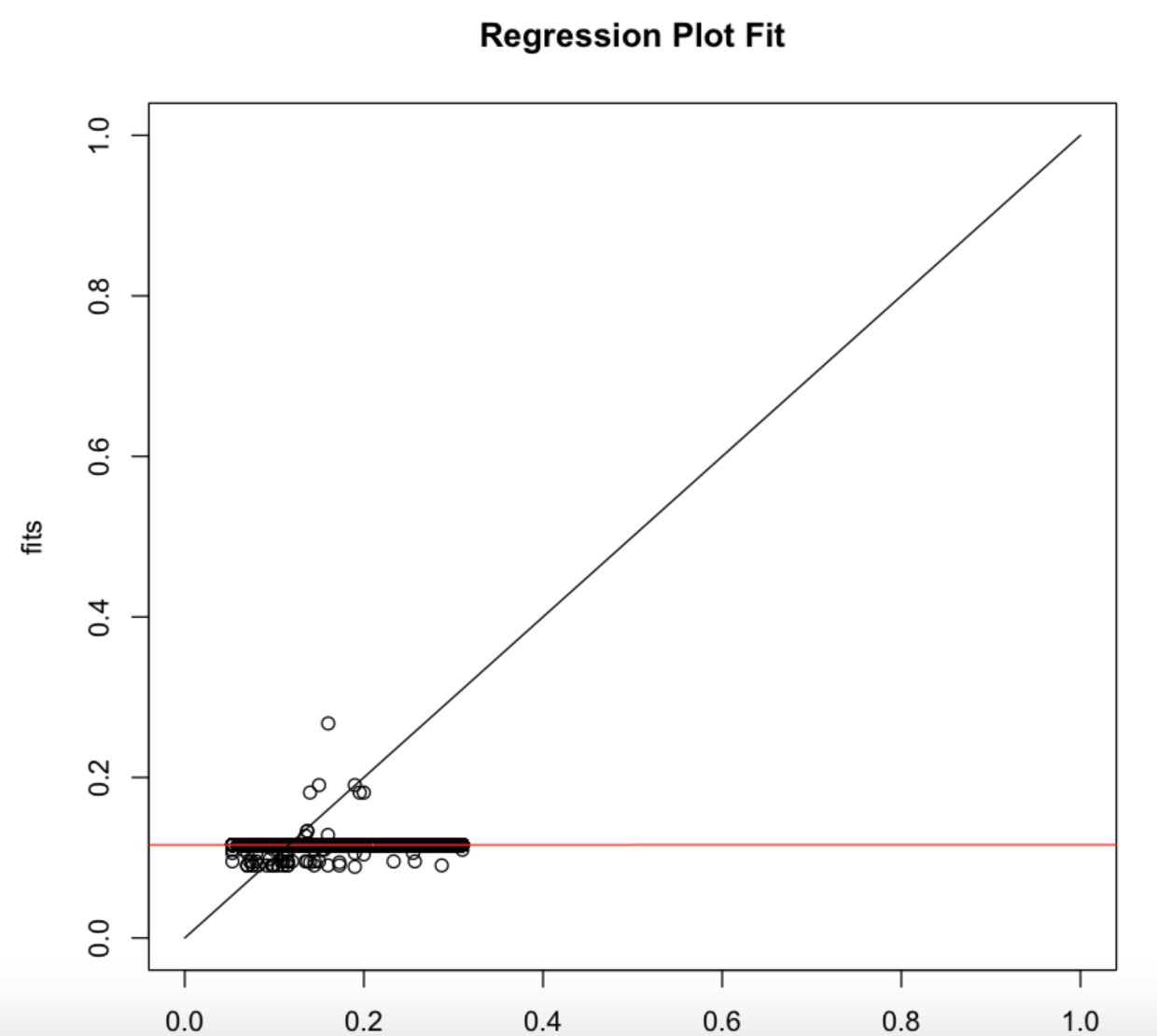


For random forest, the results are as follow.



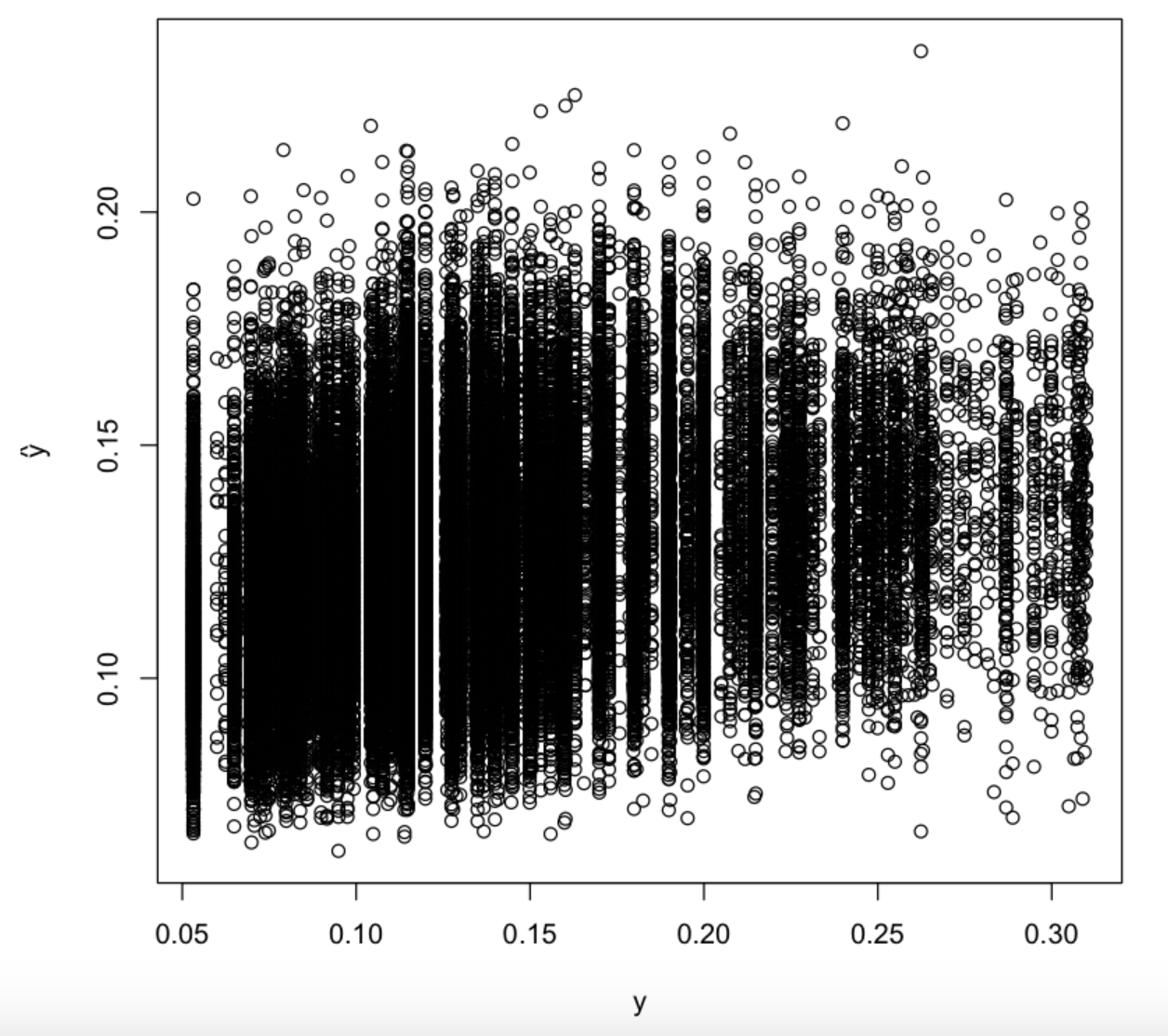
The predicted values are located closely around the real values, except for some outliers.

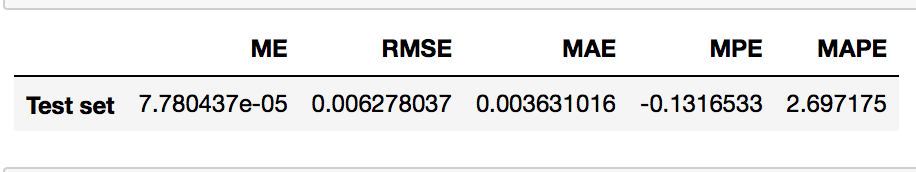
For neural network, the results are as follow.



As we can see, neural network is giving the worst result.

For KNN, we use the 10 nearest neighbors to predict. The results are as follow.

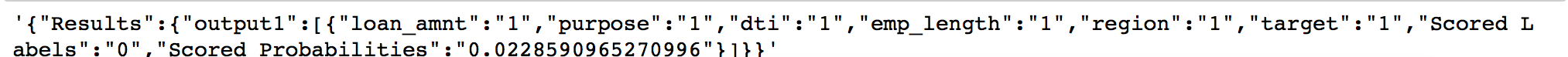




1. Deployment

We deploy the model on azure ML studio and test it with a REST API. The result is shown below.

Classification:



Prediction:

