Final Project Report

Data Mining, Spring 2020

BAD LOANS PREDICTION MODEL

Group 6

University of Texas-Arlington

1.Executive Summary

The bank loan status data set obtained from Kaggle. We use 100,000 observations and 19 variables, including 2 binary variables, 8 nominal variables, and 11 continuous variables. We use data detection, establishing a linear regression model ,decision tree model and logistic regression model on the existing basic customer information and related materials, and conduct comparative analysis to find the best model. According to the model, the customer's credit risk situation is analyzed and scored, and the best credit score to distinguish good customers from bad customers is obtained, so that credit card companies can effectively deal with a large number of credit card applicants, quickly accept Judgment of rejection, and the initial credit limit given. For customers who borrow from a bank, we can use the model to distinguish the credit rating of the customer and help the bank determine whether it should borrow from the customer. Through analysis, we found that most customers are short-term loans, mainly liquidity.

2.Project Motivation/ Background

The business question for our project is whether the customers' loan request will be approved. Based on the information of this loan customer group through this dataset, we can establish a logistic regression, a linear regression, or Decision Tree model of default customers, then we can train these three models and score it. For banks, who loan to customers, this quantitative model can be constructed to differentiate customers' credit ratings and help the bank to make a decision of whether we should loan to this customer. After knowing the default probability of each account, we can also estimate the proportion of bad debts and address some business problems in the future.

3.Data Description

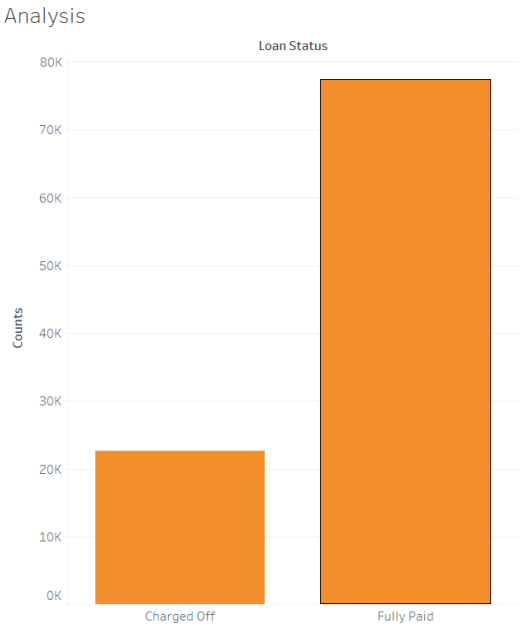
The bank loan status dataset which was obtained from Kaggle.com. The target variable Loan Status is a binary variable. And it is expressed as Full paid normal: 0, charged off overdue: 1. The goal of the target variable Loan Status in the multiple linear regression is to predict which classification the target variable ultimately belongs to : 0 or 1. In this case, we will use 0.5 (50%) as a critical value in the prediction model, which means more than 0.5 will suggest that this model is inclined to Full Paid. As for variables, there are 2 binary variables, 8 nominal variables and 11 continuous variables.

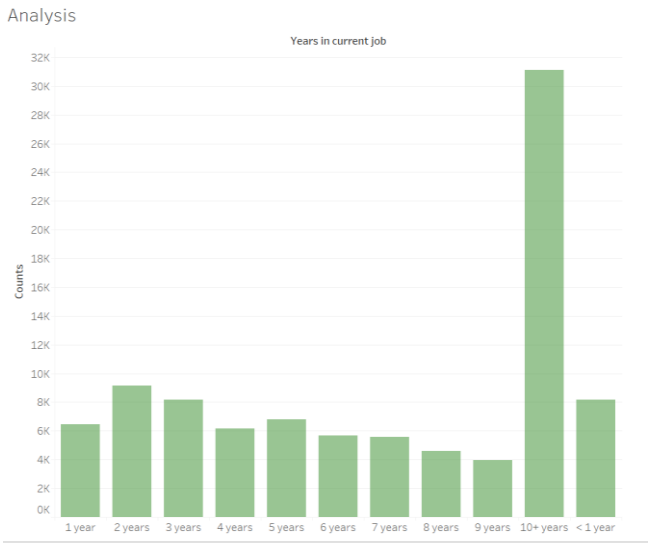
The data set is composed of 100000 observations and 19 variables:

- Loan ID
- Customer ID
- Loan Status
- Current Loan Amount
- Term
- Credit Score
- Annual Income
- Years in Current Job
- Home Ownership
- Purpose
- Monthly Debt
- Years of Credit
- History
- Month since last Delinquent
- Number of Open Accounts
- Number of Credit Problems
- Current Credit Balance
- Maximum Credit Bankruptcies
- Tax Liens

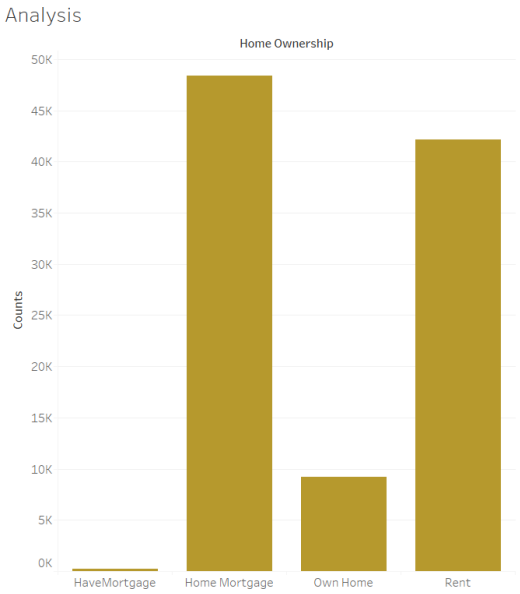
4.Data Analysis

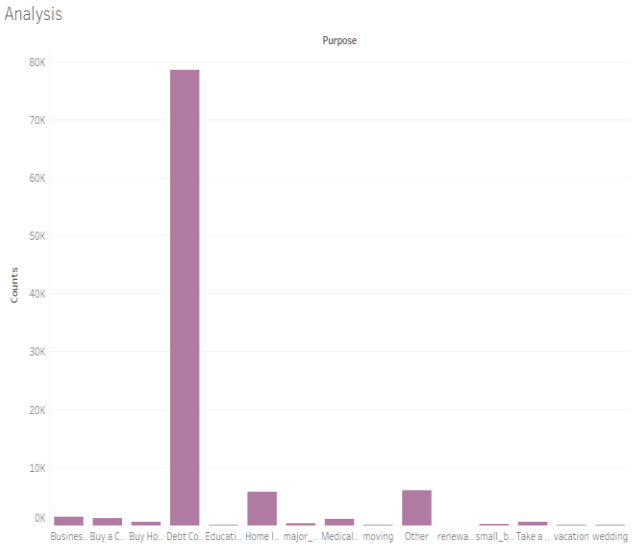
The histogram shows that the company's loan overdue situation is more serious, reaching 27.6%, and the vast majority of customers' working years are customers who have worked for more than 10 years.



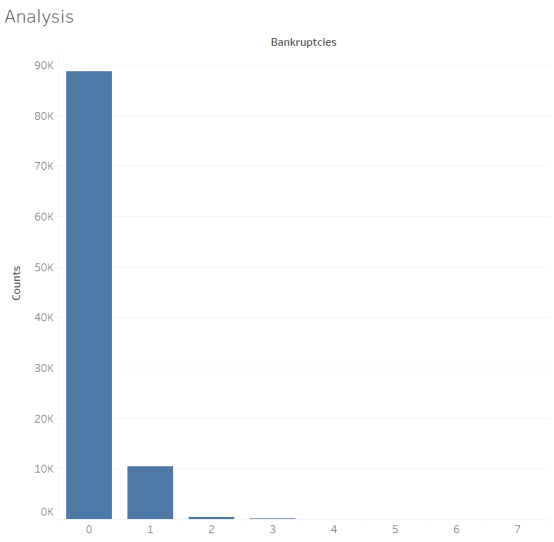


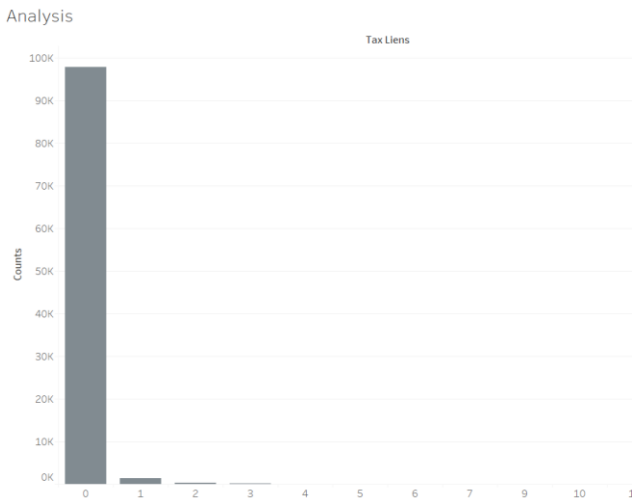
The status of Home Ownership can also be seen that the customers are mainly renting and loan customers, and debt repayment accounts for the vast majority of loan purposes, which shows that the company's asset risk is greater, which can also be explained from the one hand the reason.



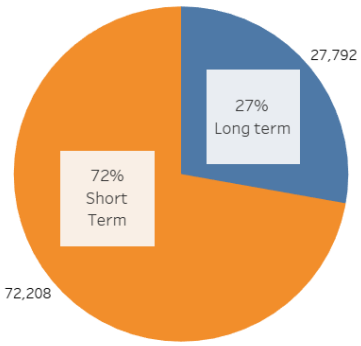


In the case of Bankruptcies and Tax Liens, we can see that the vast majority of customers have relatively good credit records, and very few customers have historically bad credit records, so we can also judge that most customers are actually the main liquidity risk , Less vicious fraud clients.



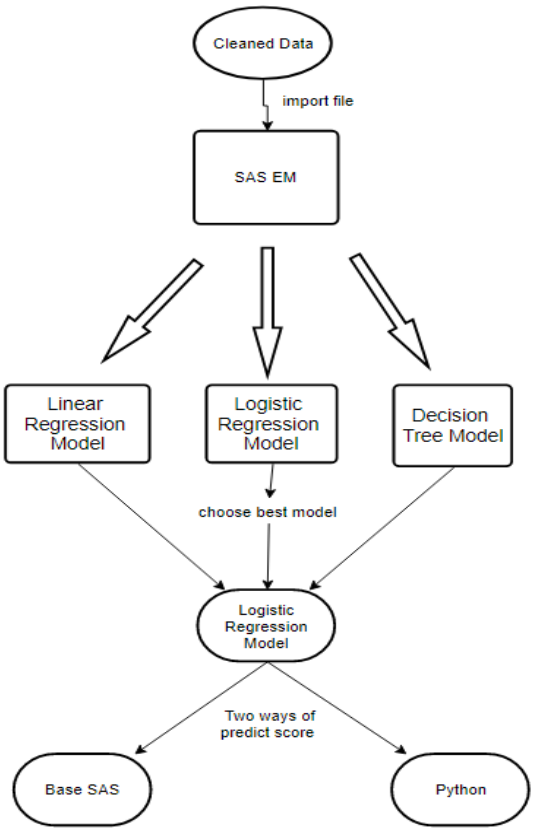


Through the pie chart, it is found that nearly 72% of the company's loan terms are short-term loans. This can explain the problem of high proportion of loan repayment in customer loan purposes from this perspective. Customer loans mainly use short-term debt repayment and provide liquidity as the Lord.

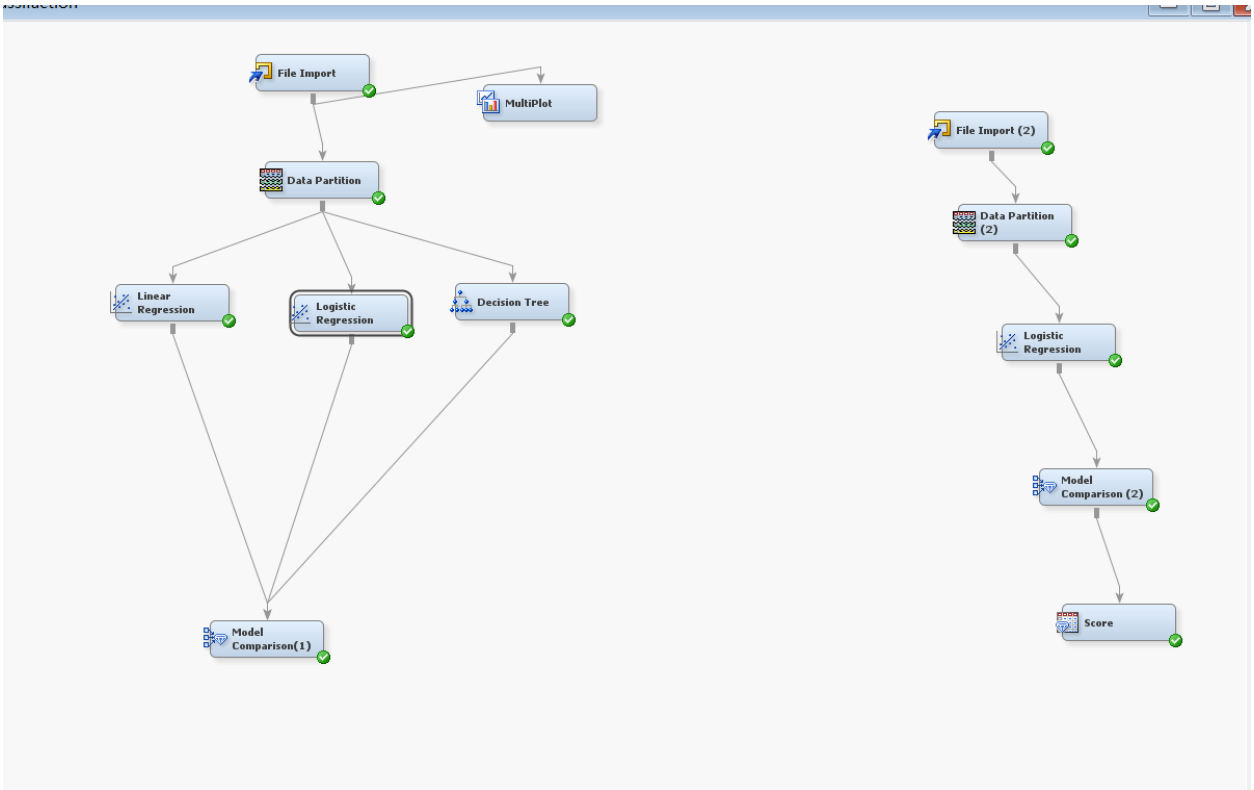


5.Prediction Models and Results

In order to verify the accuracy of the predicted data, three software are used to obtain the prediction results：1)SAS EM, 2)BASE SAS,3)PYTHON. In order to get a better prediction rate, in the latter two software BASE SAS, Python, we would compare the difference of the prediction score in the data set.



5.1 Prediction Models



5.1.1 Algorithm comparison

Algorithm	Strength	Weakness
Linear regression	The modeling speed is fast, no complicated calculations are needed, and the operation speed is still very fast in the case of a large amount of data, and the understanding and interpretation of each variable can be given according to the coefficient	Can't fit nonlinear data well. Therefore, it is necessary to determine whether there is a linear relationship between variables.
Logistic regression	Good at analyzing linear relationships, good at grasping global laws, good overall fitting effect, providing the probability or score of each observation, flexible	Sensitive to extreme values and easily affected. In order to obtain a better model effect, a large amount of data needs to be preprocessed, and the technical

	application.	requirements are high.
Decision tree	Good at analyzing non-linear relationships, can go deep into the detailed structure of data, and obtain local optimal solutions. It requires less data pre-processing and technical difficulty.	Poor grasp of the overall law is prone to overfitting.

Using the above different mining techniques to develop models, through comparative analysis, established a Logistic regression credit score model.

5.1.2 Predict Steps

First use python to clean the data, remove some missing values.

Second,import the cleaned data to the ‘import file ’node of SAS EM.

Third ,we would use a **cross-validation method** in SAS EM software,the reason is that Cross-validation can "fully utilize" limited data to find suitable model parameters and prevent overfitting .

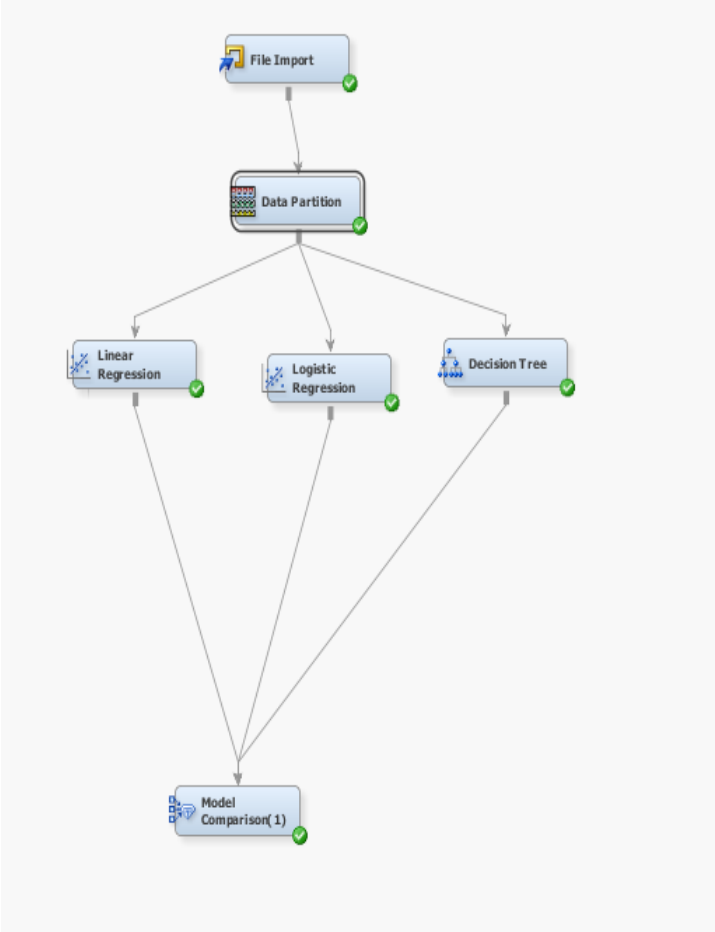
So the sample is divided into three randomly separate parts, which training set is 70%, the validation set is 20%, and the test set is 10%.

The training set is used to estimate the model, the validation set is used to determine the best model,select the model with the smallest average error rate ,and the test set is used to evaluate the performance of the best model(the accuracy rate of the best model).

5.2 Results

5.2.1 SAS EM

In the data set split, the 70% is training and the 20% is validation, use these two subsets to determine the lowest average square error on the three algorithms and choose the best.



1）The findings of the linear regression are shown below:

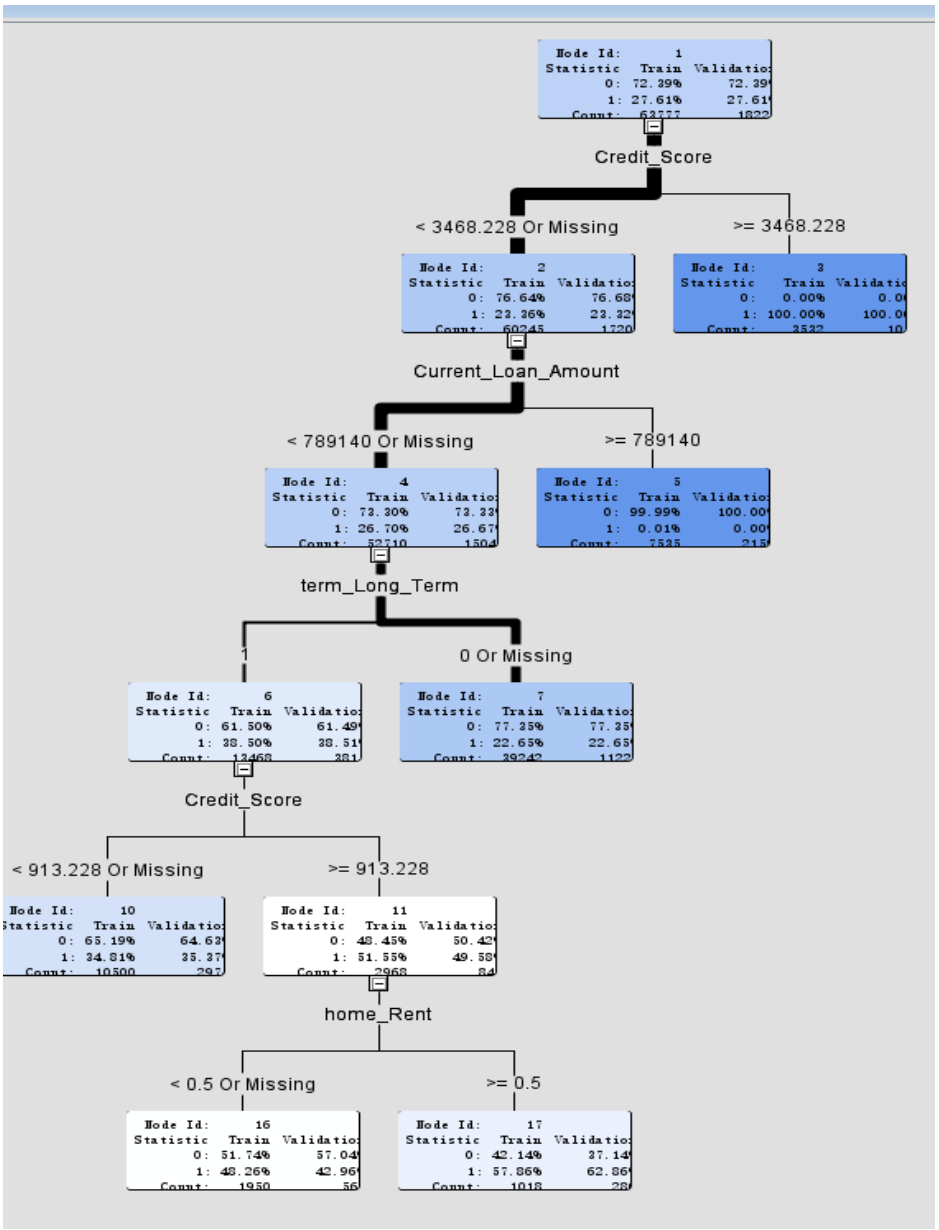
Analysis of Maximum Likelihood Estimates

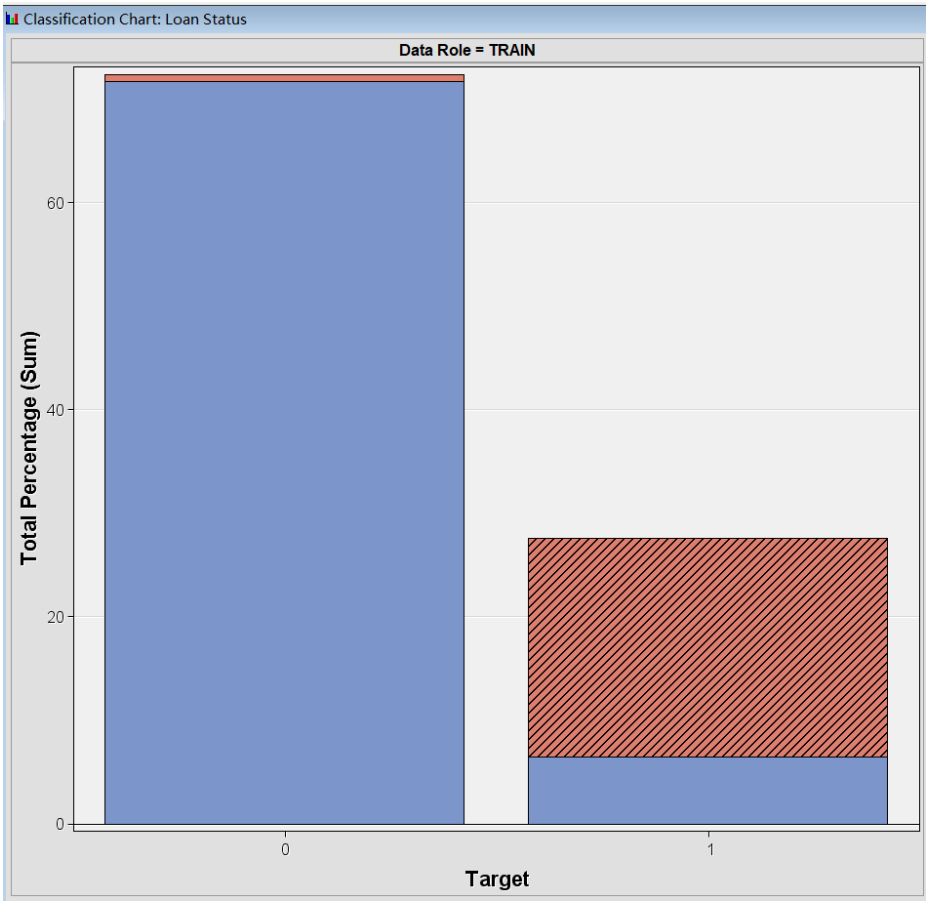
Parameter		DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept		1	0.9014	0.1469	6.14	<.0001
Annual_Income		1	2.311E-8	1.666E-9	13.87	<.0001
Credit_Score		1	-0.00011	1.085E-6	-104.28	<.0001
Current_Loan_Amount		1	2.412E-9	4.91E-11	49.17	<.0001
Monthly_Debt		1	-9.87E-7	1.558E-7	-6.34	<.0001
Number_of_Credit_Problems		1	-0.00775	0.00407	-1.90	0.0569
Number_of_Open_Accounts		1	-0.00096	0.000346	-2.76	0.0057
Tax_Liens_0_0		1	-0.1868	0.1399	-1.34	0.1818
Tax_Liens_10_0		1	0.0731	0.4212	0.17	0.8622
Tax_Liens_11_0		1	0.1146	0.4217	0.27	0.7859
Tax_Liens_15_0		1	0.1379	0.4236	0.33	0.7447
Tax_Liens_1_0		1	-0.2150	0.1407	-1.53	0.1264
Tax_Liens_2_0		1	-0.2053	0.1427	-1.44	0.1503
Tax_Liens_3_0		1	-0.2961	0.1490	-1.99	0.0469
Tax_Liens_4_0		1	-0.3154	0.1580	-2.00	0.0459
Tax_Liens_5_0		1	-0.2633	0.1817	-1.45	0.1473
Tax_Liens_6_0		1	-0.1743	0.2150	-0.81	0.4174
Tax_Liens_7_0		1	-0.3269	0.2279	-1.43	0.1514
Tax_Liens_9_0		1	0.1564	0.3149	0.50	0.6194
aim_Business_Loan		1	-0.0519	0.0463	-1.12	0.2623
aim_Buy_House		1	0.0453	0.0486	0.93	0.3506
aim_Buy_a_Car		1	0.1046	0.0466	2.25	0.0247
aim_Debt_Consolidation		1	0.0465	0.0445	1.04	0.2968
aim_Educational_Expenses		1	0.0981	0.0649	1.51	0.1307
aim_Home_Improvements		1	0.0456	0.0450	1.01	0.3102
aim_Medical_Bills		1	-0.0120	0.0470	-0.25	0.7992
aim_Other		1	0.0670	0.0453	1.48	0.1393
aim_Take_a_Trip		1	0.0138	0.0521	0.27	0.7909
aim_major_purchase		1	0.0537	0.0515	1.04	0.2972
aim_moving		1	-0.0285	0.0603	-0.47	0.6369
aim_other_1		1	0.00432	0.0450	0.10	0.9234
aim_renewable_energy		1	-0.0197	0.1559	-0.13	0.8995
aim_small_business		1	-0.1634	0.0532	-3.07	0.0021
aim_vacation		1	-0.0211	0.0684	-0.31	0.7578
aim_wedding		0	0	.	.	.
home_HaveMortgage		1	0.1148	0.0443	2.60	0.0095
home_Home_Mortgage		1	0.0562	0.00345	16.29	<.0001
home_Own_Home		1	0.0284	0.00577	4.91	<.0001
home_Rent		0	0	.	.	.
term_Long_Term	0	1	0.0729	0.00184	39.62	<.0001
term_Short_Term	0	0	0	.	.	.

From above result, we can see that Annual_ income ,Credit_Score,Current_Loan_Amount,

Monthly_Dept,home_Home_Mortgage,home_Own_Home,term_Long_Term are significant to our target variable churn, since they are associated with very small P values.

2）The findings of the decision tree are shown below:





Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
Loan_Status	Loan Status	_NOBS_	Sum of Frequencies	63777	18222	
Loan_Status	Loan Status	_MISC_	Misclassification ...	0.218198	0.216222	
Loan_Status	Loan Status	_MAX_	Maximum Absolut...	0.999867	0.773457	
Loan_Status	Loan Status	_SSE_	Sum of Squared E...	19989.73	5704.353	
Loan_Status	Loan Status	_ASE_	Average Squared ...	0.156716	0.156524	
Loan_Status	Loan Status	_RASE_	Root Average Squ...	0.395873	0.395631	
Loan_Status	Loan Status	_DIV_	Divisor for ASE	127554	36444	
Loan_Status	Loan Status	_DFT_	Total Degrees of ...	63777	.	

Classification Table

Data Role=TRAIN Target Variable=Loan_Status Target Label=Loan Status

Targe	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	77.2283	99.0708	45740	71.7186
1	0	22.7717	76.5959	13487	21.1471
0	1	9.4286	0.9292	429	0.6727
1	1	90.5714	23.4041	4121	6.4616

It can be seen from the output graph of the model that when a decision tree with 17 leaf nodes is generated, the misclassification rate of the training set is 21.8%, and the misclassification rate of the validation set is 21.6%. In the training set, 46,740 of "good customers"(Full paid normal: 0) have 45,740 correctly classified, and 17,608 of "bad customers"(charged off overdue: 1) have 4,121 correctly classified.

3) Since the target variable Loan_stauts is binary, a logistic regression was run in SAS EM using stepwise method:

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Te
Loan_Status	Loan Status	_AIC_	Akaike's Information Criterion	58898.85	.	
Loan_Status	Loan Status	_ASE_	Average Squared Error	0.15425	0.153513	
Loan_Status	Loan Status	_AVERR_	Average Error Function	0.461521	0.459959	
Loan_Status	Loan Status	_DFE_	Degrees of Freedom for Error	63762	.	
Loan_Status	Loan Status	_DFM_	Model Degrees of Freedom	15	.	
Loan_Status	Loan Status	_DFT_	Total Degrees of Freedom	63777	.	
Loan_Status	Loan Status	_DIV_	Divisor for ASE	127554	36444	
Loan_Status	Loan Status	_ERR_	Error Function	58868.85	16762.76	
Loan_Status	Loan Status	_FPE_	Final Prediction Error	0.154322	.	
Loan_Status	Loan Status	_MAX_	Maximum Absolute Error	1	0.999922	
Loan_Status	Loan Status	_MSE_	Mean Square Error	0.154286	0.153513	
Loan_Status	Loan Status	_NOBS_	Sum of Frequencies	63777	18222	
Loan_Status	Loan Status	_NW_	Number of Estimate Weights	15	.	
Loan_Status	Loan Status	_RASE_	Root Average Sum of Squar...	0.392746	0.391807	
Loan_Status	Loan Status	_RFPE_	Root Final Prediction Error	0.392839	.	
Loan_Status	Loan Status	_RMSE_	Root Mean Squared Error	0.392792	0.391807	
Loan_Status	Loan Status	_SBC_	Schwarz's Bayesian Criterion	59034.8	.	
Loan_Status	Loan Status	_SSE_	Sum of Squared Errors	19675.16	5594.611	
Loan_Status	Loan Status	_SUMW_	Sum of Case Weights Time	127554	36444	
Loan_Status	Loan Status	_MISC_	Misclassification Rate	0.217492	0.217484	
Loan_Status	Loan Status	_XMISC_	Misclassification Rate	0.217571	.	

Data Role=TRAIN Target Variable=Loan_Status Target Label=Loan Status

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	77.3239	98.9842	45700	71.6559
1	0	22.6761	76.1131	13402	21.0138
0	1	10.0321	1.0158	469	0.7354
1	1	89.9679	23.8869	4206	6.5949

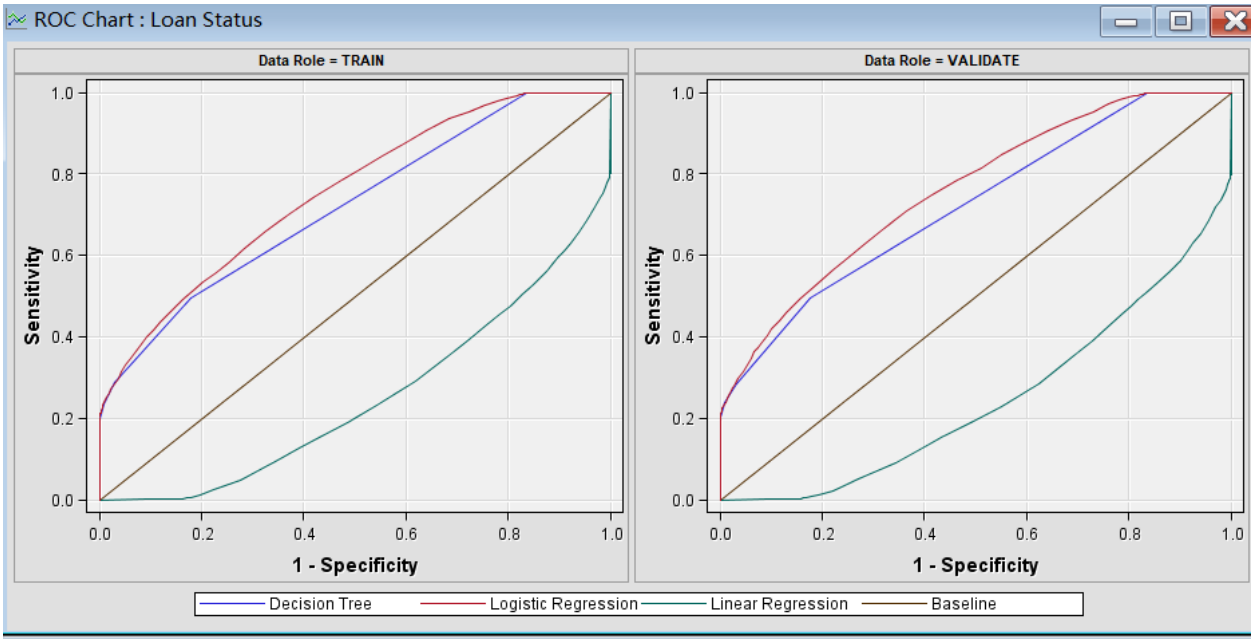
It can be seen from the above training set graph that the overall model accuracy rate is 1-0.21 = 79%, and the model predicts 98.98% of good customers(Full paid normal: 0) as good customers and 1% of good customers as bad customers. 76% of bad customers(charged off overdue: 1) are predicted to be good customers, and the remaining 23% of bad customers are predicted to be bad customers.

5.2.1.1Which one is the best model

In three models: the smallest average squared error rate is 0.15347=15%, the best model is the Logistic Regression Model.

Fit Statistics
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	Tree	Decision Tree	0.21622	0.15672	0.21820	0.15652
	Reg2	Logistic Regression	0.21787	0.15421	0.21749	0.15347
	Reg	Linear Regression	0.21957	0.15611	0.22020	0.15544



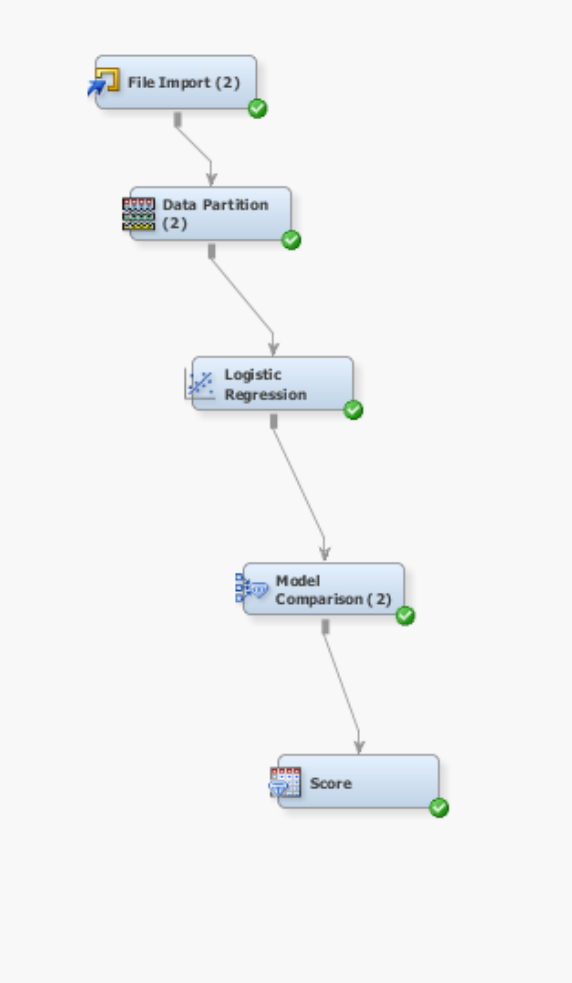
In addition ,the ROC curve also showed the performance of the model.

From the above plot,better performance can be reflected by the curve near the upper left corner.

So on the other hand, near the upper left corner is the logistic regression model(red line), from which we can know that the logistic regression model is the best model in prediction.

5.2.1.2 Accuracy Measures in SAS EM:

In test set: Use the logistic regression model on the training set (90%)to predict the last 10% test



The accuracy rate of test set in logistic regression model-Model Comparison(2):

Note: **Loan status (0)** is expressed as Full paid, **Loan status (1)** is expressed as charged off, because the target variable is loan status, so the accuracy rate is clearly expressed here as the correct rate of predicting the passing loan.

Data Role=Test

Statistics	Reg3
Test: Kolmogorov-Smirnov Statistic	0.36
Test: Average Squared Error	0.15
Test: Roc Index	0.76
Test: Average Error Function	0.46
Test: Bin-Based Two-Way Kolmogorov-Smirnov Probability Cutoff	0.29
Test: Cumulative Percent Captured Response	29.20
Test: Percent Captured Response	11.04
Test: Divisor for TASE	16402.00
Test: Error Function	7471.33
Test: Gain	191.64
Test: Gini Coefficient	0.53
Test: Bin-Based Two-Way Kolmogorov-Smirnov Statistic	0.35
Test: Kolmogorov-Smirnov Probability Cutoff	0.27
Test: Cumulative Lift	2.92
Test: Lift	2.21
Test: Maximum Absolute Error	1.00
Test: Misclassification Rate	0.21
Test: Lower 95% Conf. Limit for TMISC	0.20
Test: Upper 95% Conf. Limit for TMISC	0.22
Test: Mean Square Error	0.15
Test: Sum of Frequencies	8201.00
Test: Root Average Squared Error	0.39
Test: Cumulative Percent Response	80.51
Test: Percent Response	60.98
Test: Root Mean Square Error	0.39
Test: Sum of Square Errors	2488.03
Test: Sum of Case Weights Times Freq	16402.00

Accuracy rate=(TP+TN)/(TP+FN+FP+TN)=1-Misclassification Rate=1-21%=79%

The logistic regression model predicted 79% of correct results, this means that the loan classifier performs very well in identifying fraudulent customers.

5.2.2 Base SAS(Predict Score1)

Note:
P_Loan_Status1 Predicted: Loan_Status=1

P_Loan_Status0 Predicted: Loan_Status=0

To get the Loan Status ' predicted probability, use the optimized SAS code from the score node, save the SAS optimized code in the same file path and run it in Base SAS.

The predicted score of Loan Status(0) and Loan Status(1):(pick up 10 observations)

```
data myPredicted;
      set Work.Scorereglogistic;
length _WARN_ $4;
label _WARN_ = 'Warnings' ;

length I_Loan_Status $ 12;
label I_Loan_Status = 'Into: Loan_Status' ;
*** Target Values;
array REG3DRF [2] $12 _temporary_ ('1' '0' );
label U_Loan_Status = 'Unnormalized Into: Loan_Status' ;
*** Unnormalized target values;
ARRAY REG3DRU[2]  _TEMPORARY_ (1 0);

drop _DM_BAD;
_DM_BAD=0;

*** Check Annual_Income for missing values ;
if missing( Annual_Income ) then do;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
```

```
* TYPE: ASSESS;
* NODE: Score;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* Score: Creating Fixed Names;
*-------------------------------------------------------------*;
LABEL EM_SEGMENT = 'Segment';
EM_SEGMENT = b_Loan_Status;
LABEL EM_EVENTPROBABILITY = 'Probability for level 1 of Loan_Status';
EM_EVENTPROBABILITY = P_Loan_Status1;
LABEL EM_PROBABILITY = 'Probability of Classification';
EM_PROBABILITY =
max(
P_Loan_Status1
,
P_Loan_Status0
);
LENGTH EM_CLASSIFICATION $%dmnorlen;
LABEL EM_CLASSIFICATION = "Prediction for Loan_Status";
EM_CLASSIFICATION = I_Loan_Status;
PROC SORT DATA=myPredicted;
      BY DESCENDING P_Loan_Status0;
run;
title "The First 10 Observations";
PROC PRINT DATA=myPredicted(obs=10);
        P_Loan_Status1 P_Loan_Status0;
run;
```

The First 10 Observations

Obs	P_Loan_Status1	P_Loan_Status0
1	0.27609	0.72391
2	0.27609	0.72391
3	0.27609	0.72391
4	0.27609	0.72391
5	0.27609	0.72391
6	0.27609	0.72391
7	0.27609	0.72391
8	0.27609	0.72391
9	0.27609	0.72391
10	0.27609	0.72391

When we move to the probability of classification，we notice that the first 10 observations have the same probability in both status 0 and 1. Which means they have 72% of probability to be

fully-paid, and 27% to be charged off. which means the customer's credit score is good and the bank can lend to the customer.

5.2.3Python-(Predict Score2)

To get the Loan Status predicted score:78%

Train the model and give a score, the accuracy rate is 78.88%, and the prediction success rate is high.

```
In [16]: from sklearn.model_selection import train_test_split # split the data set
         from sklearn.linear_model import LogisticRegression

         num= loandf.shape[0]
         source_X=newloandf.loc[0:num-1,:]
         source_y=newloandf.loc[0:num-1,'Loan Status']

         train_X,test_X,train_Y,test_y= train_test_split(source_X,source_y,test_size=.1,random_state = 0)#10%test



         model=LogisticRegression()
         model.fit(source_X,source_y)
         model.score(test_X,test_y)

Out[16]: 0.7888109886965231
```

6.Conclusions

Summary	Model Accuracy Rate
SAS EM-Logistic Regression	79%
BASE-SAS	72%
Python	78.888%

The reason why we use three software to score customers is to verify whether the logistic regression models obtained by these three software have too much deviation. As can be seen from the above figure, the deviation is not big**,** the error is very small, only from 0.1% to 0.7%. The accuracy rate of the model is between 72% -79%. The reason why the BASE SAS software score is the smallest is because we took the average of 10 observations and selected the average model accuracy rate.

To the most important, based on the critical value of 0.5, the value of 70% is greater than 0.5. It can be inferred that the bank can lend to the customer and the customer has good credit. The customers' loan request will be approved.

Our experiences and suggestions:

In this project we learned a lot, applied the knowledge of this class, and learned how to use different software to score credit cards.

Credit scoring is actually the application of scoring technology in credit risk management. The establishment of a credit score is based on the statistical results of a large number of data, which has very high accuracy and reliability. The application credit score is used exclusively for the credit evaluation of new applicants. It can quickly and effectively identify and divide the advantages and disadvantages of customers through the relevant identity information filled out

by the applicant, prevent customers with bad credit from applying for credit cards, improve the credit level of cardholders, realize the precaution of credit card business risks, and help banks establish The first credit risk safety net.

However, the credit card application scoring model designed in this paper cannot replace the credit scoring system. Only with the support of the credit scoring system will it be possible to obtain rich and complete customer information. With the development of the credit card business and the credit status of the entire society, the key factors that determine customer credit will also continue to change. Therefore, data mining methods can Flexible use. By discovering the rules in the customer information, finding out the elements that the bank needs to focus on, it plays a directional guiding role for the verification and verification of the bank. In this way, both efficiency and efficiency are guaranteed. In addition, some new models discovered through data mining can further adjust the customer credit scoring system, thereby playing an important role in improving the credit scoring system in the future. The credit card scoring table model constructed in this paper uses a variety of data modeling and mining techniques, which can effectively, objectively, accurately and consistently make effective credit evaluations for credit card applicants, which has a certain auxiliary effect on the actual credit card application scoring business.

Due to the short research time, coupled with the limitations of own knowledge, information, experience, and energy, this credit scoring model still has some limitations and needs to be developed, mainly in the following aspects because this research involves some commercial banks Secretly, the data samples obtained are very limited. Even the data obtained has problems in all aspects of data quality. Many key data have been fuzzified, and at the same time, the missing values are serious, which leads to a decrease in the accuracy of the scoring model. The

data obtained in this research is based on the customer data that has been scored by bank credit staff, but the credit staff itself has certain subjective factors, and there are certain errors and inconsistencies in the classification of customers, so this model will inevitably produce this aspect error.

Since the United States has a very sound personal credit system, the data related to personal credit is distributed among various functional departments and related units, and between various departments and banks. In this situation, the credit risk assessment of the credit card mainly relies on the data characteristics of the customer, which is used to evaluate the customer's credit status. Banks may still face some credit risks in the process of credit card approval. This credit card scoring model is only a preliminary attempt to use data mining techniques to analyze the customer data of the issuing bank to assist the risk management of the issuing bank. With the development of the American personal credit system, the improvement of the corresponding legal environment, the development of informatization, the realization of information resource sharing, the strengthening of the national personal credit concept and the recognition of credit, data mining technology will become the bank ’s credit risk management, customers An important tool for relationship management, financial product development, marketing decision analysis, and improved banking management.

Appendix

Python Code:

```
# Input data to do data cleansing

import pandas as pd
import numpy as ny
import matplotlib.pyplot as plt
import seaborn as sns

loandf = pd.read_csv("E:/INSY 专业/Spring2020/INSY 5339/my-
dataset/credit_train.csv")
print(" The data size:",loandf.shape)
loandf.head



lossdf=loandf.isnull().sum().reset_index()
lossdf['Missing the percentage'] = (lossdf[0]/len(loandf))*100
lossdf.columns=['Column name','Missing Value','Missing percentage']
lossdf = lossdf[lossdf['Missing percentage']!=0].sort_values('Missing
percentage',ascending=False)
lossdf


#Missingvaluehandling.Deletemissingvalues


nadf = loandf[loandf['Loan ID'].isnull()==True].index
loandf = loandf.drop(nadf,axis=0)
loandf.info()


# In[5]:


lossdf=loandf.isnull().sum().reset_index()
lossdf['Missing the percentage'] = (lossdf[0]/len(loandf))*100
lossdf.columns=['Column name','missing value','Missing percentage']
lossdf = lossdf[lossdf['Missing percentage']!=0].sort_values('Missing
percentage',ascending=False)
lossdf


# In[4]:


loandf.drop(columns='Months since last delinquent',inplace=True,axis=1)
loandf.info()
```

```
# In[5]:


loandf['Credit Score']=loandf['Credit Score'].fillna(loandf['Credit
Score'].mean())
loandf['Annual Income']=loandf['Annual Income'].fillna(loandf['Annual
Income'].mean())
loandf['Maximum Open Credit']=loandf['Maximum Open
Credit'].fillna(loandf['Maximum Open Credit'].mean())

loandf.info()


# In[6]:


from collections import Counter
print(Counter(loandf['Years in current job']).most_common(1))
print(Counter(loandf['Bankruptcies']).most_common(1))
print(Counter(loandf['Tax Liens']).most_common(1))


# In[7]:


loandf['Years in current job']=loandf['Years in current job'].fillna('10+
years')
loandf['Bankruptcies']=loandf['Bankruptcies'].fillna('0')
loandf['Tax Liens']=loandf['Tax Liens'].fillna('0')
loandf.info()


#  Duplicate value deletion


loandf[loandf.duplicated()].count()
loandf.drop_duplicates('Loan ID', 'first',inplace=True)
loandf.info()


#Feature extraction


yeardic= {'1 years':'1-3',
          '2 years': '1-3',
          '3 years':'1-3',
          '4 years':'4-6',
          '5 years':'4-6',
          '6 years':'4-6',
          '7 years':'7-9',
          '8 years':'7-9',
          '9 years':'7-9',
          '10+ years':'10+'
}
```

```
loandf['Years in current job']=loandf['Years in current job'].map(yeardic)
loandf.head()

yeardf = pd.DataFrame()
yeardf = pd.get_dummies(loandf['Years in current job'])
loandf = pd.concat([loandf,yeardf],axis=1)
loandf.head()


# In[9]:


stdf=loandf['Loan Status'].map(lambda a :0 if a =='Fully Paid' else 1)
stdf.columns=['Status']
loandf.drop('Loan Status',inplace=True, axis=1)
loandf=pd.concat([loandf,stdf],axis=1)
loandf.head()


# In[10]:


termdf = pd.get_dummies(loandf['Term'],prefix='term')
loandf=pd.concat([loandf,termdf],axis=1)
loandf.head()


# In[11]:


homedf = pd.get_dummies(loandf['Home Ownership'],prefix='home')
loandf=pd.concat([loandf,homedf],axis=1)
loandf.head()


# In[12]:


aimdf=pd.get_dummies(loandf['Purpose'],prefix='aim')
loandf=pd.concat([loandf,aimdf],axis=1)
loandf.head()


# In[13]:


brdf=pd.get_dummies(loandf['Bankruptcies'],prefix='Bankruptcies')
loandf=pd.concat([loandf,brdf],axis=1)
loandf.head()


# In[14]:


tldf=pd.get_dummies(loandf['Tax Liens'],prefix='Tax Liens')
```

```
loandf=pd.concat([loandf,tldf],axis=1)
loandf.head()


# Obtain correlation coefficient:


corrdf=loandf.corr()
corrdf['Loan Status'].sort_values(ascending =False)


# Get the cleaned data::

newloandf = pd.concat([loandf['Credit
Score'],termdf,homedf,aimdf,loandf['Monthly Debt'],loandf['Number of Open
Accounts'],
                       loandf['Number of Credit
Problems'],tldf,loandf['Annual Income'],
                       loandf['Current Loan Amount'],loandf['Loan
Status']],axis=1)

newloandf.head()
#newloandf.to_excel("1.xlsx")


# Train the model:


from sklearn.model_selection import train_test_split # split the data set
from sklearn.linear_model import LogisticRegression

num= loandf.shape[0]
source_X=newloandf.loc[0:num-1,:]
source_y=newloandf.loc[0:num-1,'Loan Status']

train_X,test_X,train_Y,test_y=
train_test_split(source_X,source_y,test_size=.1,random_state = 0)#10%test


# Score the model

model=LogisticRegression()
model.fit(source_X,source_y)
model.score(test_X,test_y)
```

References

1. Data Mining Using SAS® Enterprise Miner™: A Case Study Approach, Fourth Edition https://documentation.sas.com/?docsetId=emcs&docsetTarget=p0918cip9ec6krn1dn1setm8vk8l.htm&docsetVersion=14.3&locale=en#p1xyk6agg32v1on1xkfnnos0bvj9
2. Textbook- Data Mining for Business Analytics. Concepts, Techniques and Applications in R ISBN-10: 1118879368