Credit Card Delinquency Prediction using ML

By Atul Pai

What is Credit Delinquency?

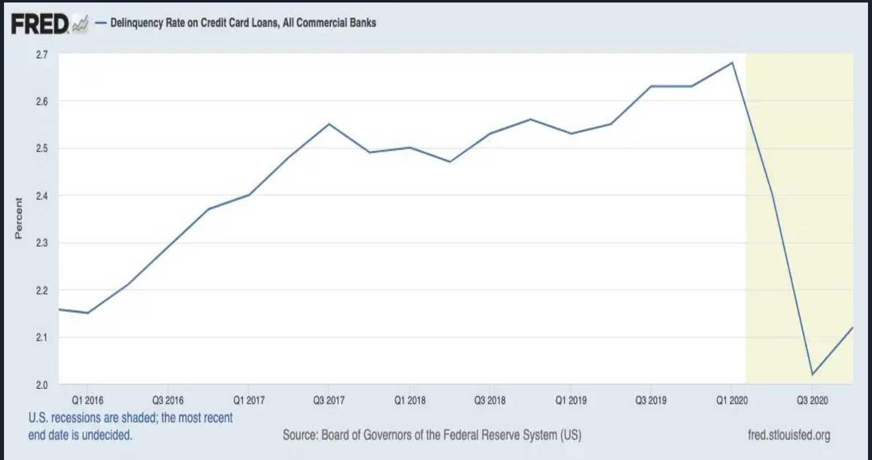
Delinquency vs Default:

Delinquency:

In general is a slightly mild term where a borrower is behind on payments. (usually 60 days)

Default :

Once a person or institution is delinquent for over a specified period (usually 9 months).



Why does it matter to Financial Institutions? What do they do?

Reasons to Monitor:

- Macro Factors : Loan delinquencies which eventually lead to default can cause Global Financial Crisis.
- Reputation: Analysts measure a bank based on their Delinquency rates, affects their appearance in the eyes of their shareholders.
- Operational Costs : Bear collection and loss of capital costs .

Factors generally used to predict:

- Credit Score/FICO Score
- Interest Rate
- Debt-to-income ratio of the borrower
- Number of days with a credit line
- Borrower’s revolving balance
- Utilization rate of the borrower’s revolving line
- Number of times the borrower had not paid in full or gone 30+ days past the due date on payment in the last two years.

Dataset :
Kaggle Competition : Give Me Some Credit

Variable Name	Description	Type
SeriousDlqin2yrs	**Person experienced 90 days past due delinquency or worse**	**Y/N**
RevolvingUtilizationOfUnsecuredLines	Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits	percentage
age	Age of borrower in years	integer
NumberOfTime30-59DaysPastDueNotWorse	Number of times borrower has been 30-59 days past due but no worse in the last 2 years.	integer
DebtRatio	Monthly debt payments, alimony,living costs divided by monthy gross income	percentage
MonthlyIncome	Monthly income	real
NumberOfOpenCreditLinesAndLoans	Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards)	integer
NumberOfTimes90DaysLate	Number of times borrower has been 90 days or more past due.	integer
NumberRealEstateLoansOrLines	Number of mortgage and real estate loans including home equity lines of credit	integer
NumberOfTime60-89DaysPastDueNotWorse	Number of times borrower has been 60-89 days past due but no worse in the last 2 years.	integer
NumberOfDependents	Number of dependents in family excluding themselves (spouse, children etc.)	integer

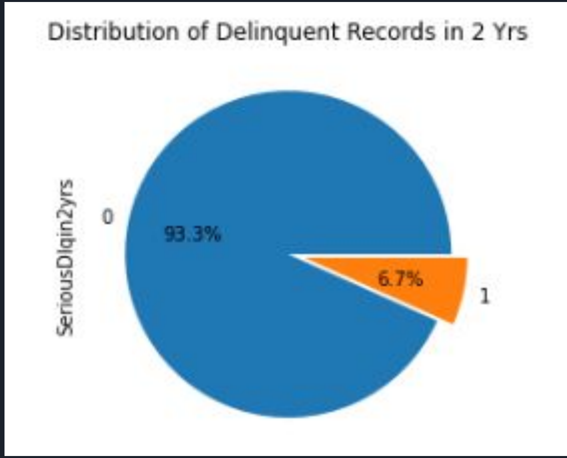
Problem:

Based on the above features predict if an applicant will be delinquent in the next 2 years.

Number of Records : 150000

Exploratory Data Analysis:

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	150000 non-null	int64
1	SeriousDlqin2yrs	150000 non-null	int64
2	RevolvingUtilizationOfUnsecuredLines	150000 non-null	float64
3	age	150000 non-null	int64
4	NumberOfTime30-59DaysPastDueNotWorse	150000 non-null	int64
5	DebtRatio	150000 non-null	float64
6	MonthlyIncome	120269 non-null	float64
7	NumberOfOpenCreditLinesAndLoans	150000 non-null	int64
8	NumberOfTimes90DaysLate	150000 non-null	int64
9	NumberRealEstateLoansOrLines	150000 non-null	int64
10	NumberOfTime60-89DaysPastDueNotWorse	150000 non-null	int64
11	NumberOfDependents	146076 non-null	float64

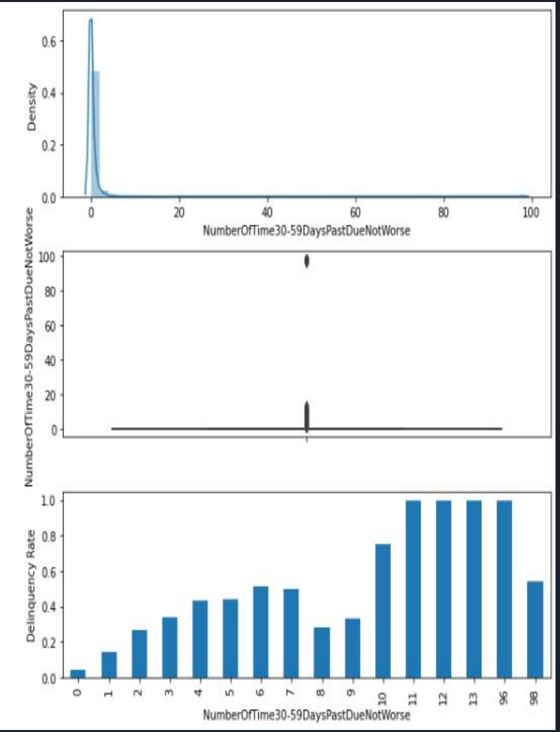


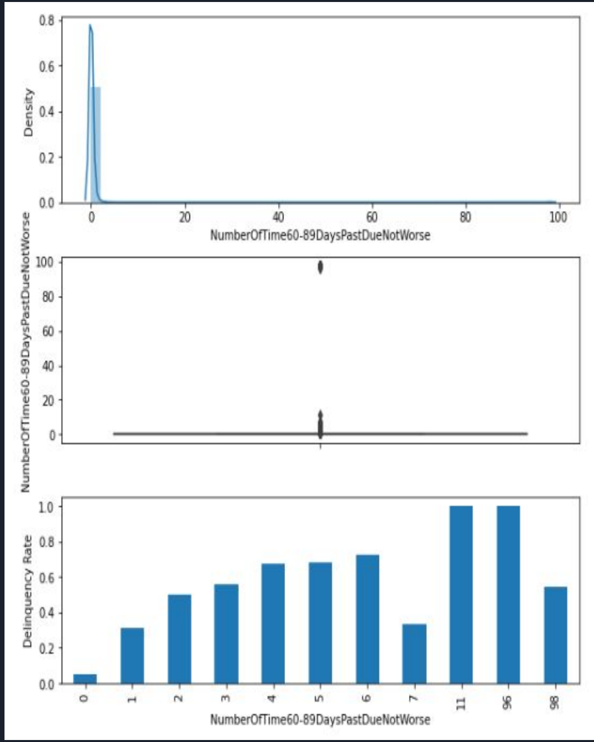
- Highly Imbalanced Dataset
- Null Values

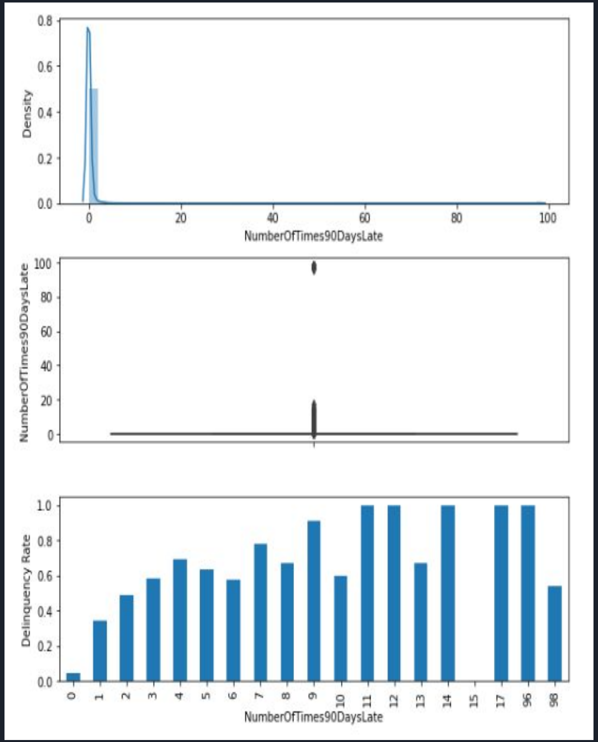
Exploratory Data Analysis: Univariate Analysis

See how the target variable changes with each of the explanatory variables:

Frequency Features:



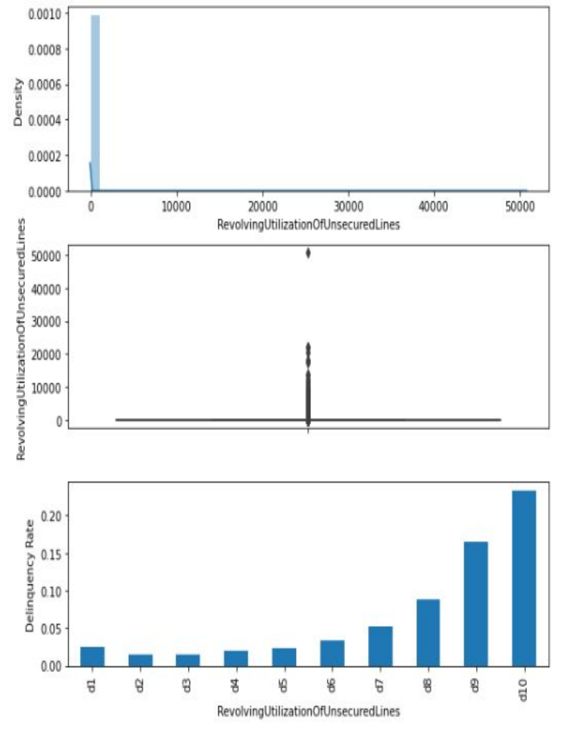


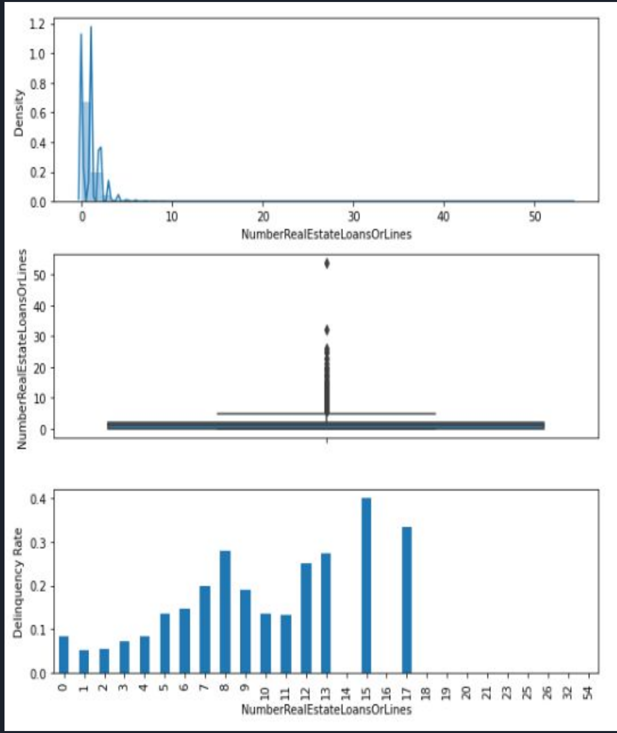


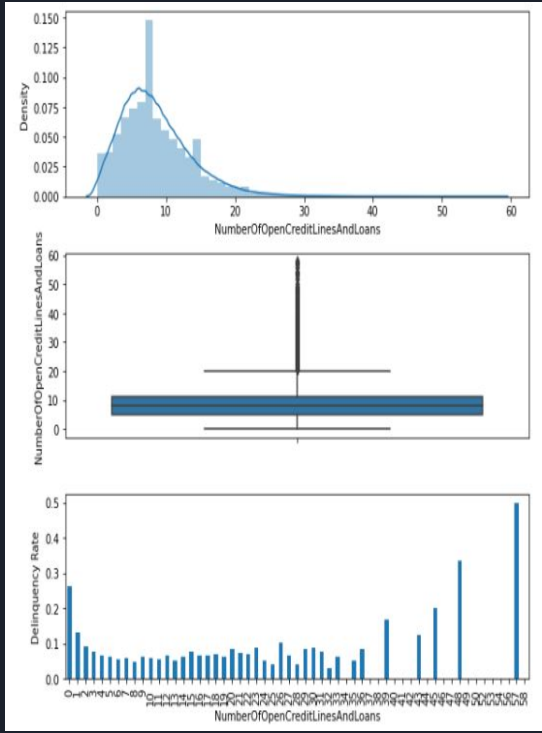
Exploratory Data Analysis: Univariate Analysis

See how the target variable changes with each of the explanatory variables:

Revolving Utilization, Number of Real Estate Loans, Number of Credit Lines



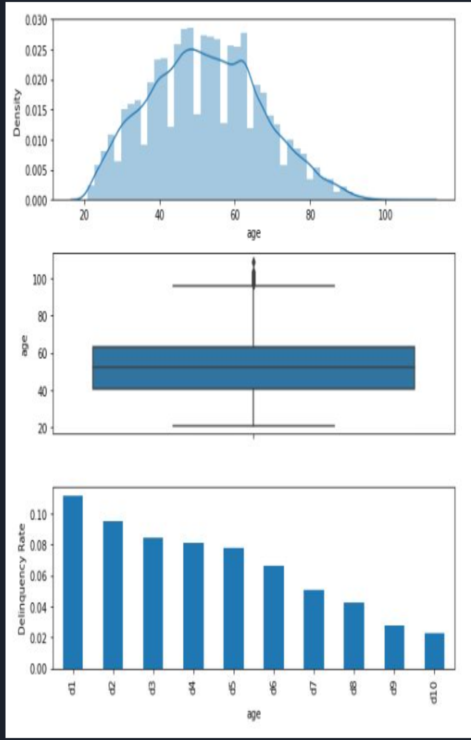


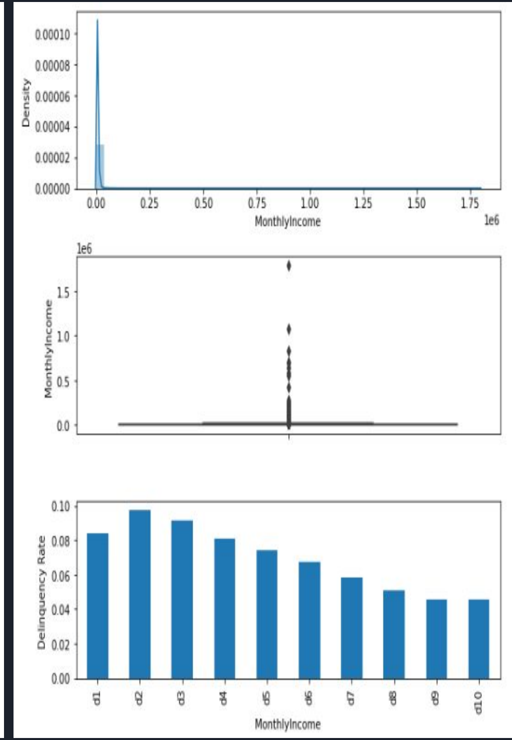


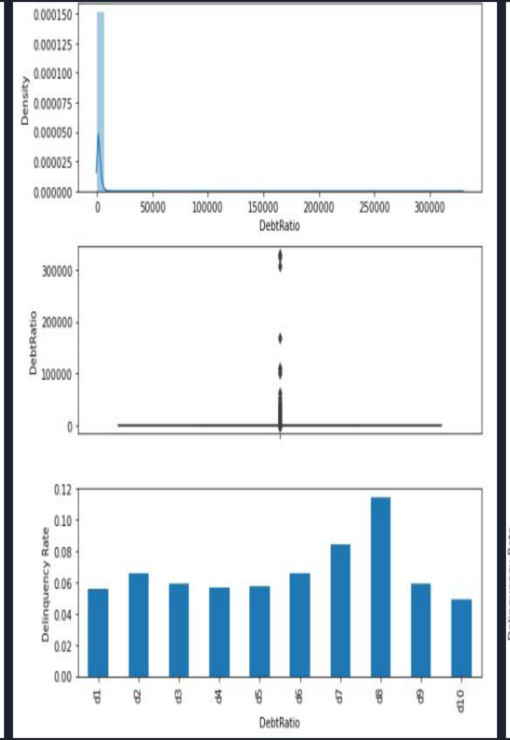
Exploratory Data Analysis: Univariate Analysis

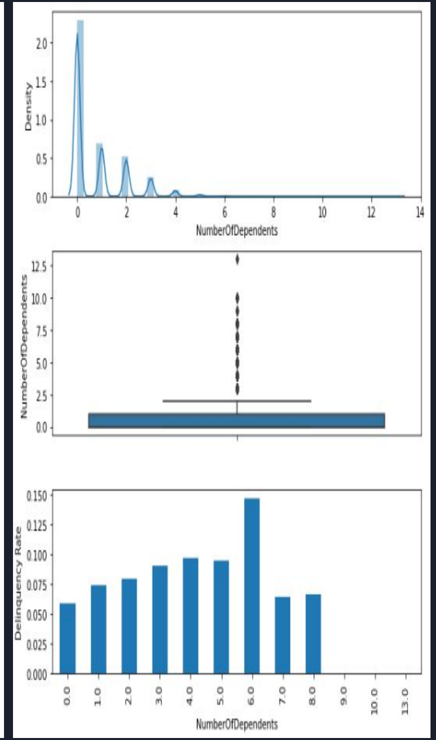
See how the target variable changes with each of the explanatory variables:

Age, Monthly Income, Debt Ratio, Number of Dependents









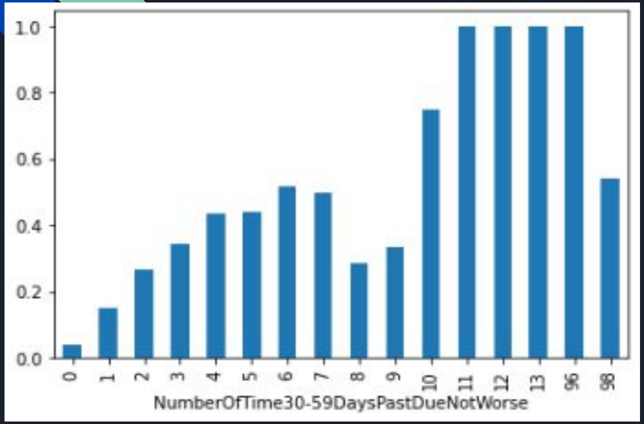
Exploratory Data Analysis: Univariate Analysis

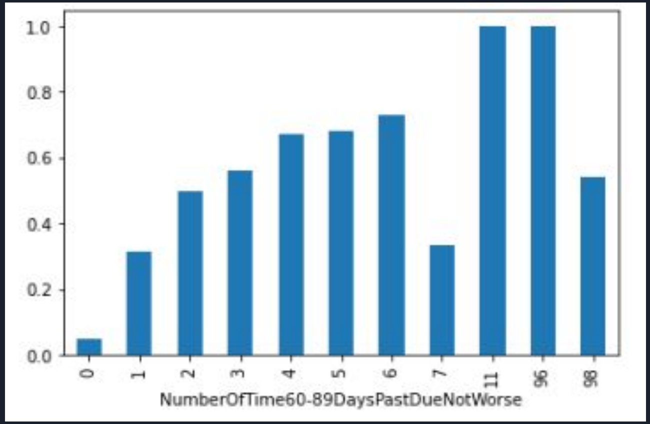
See how the target variable changes with each of the explanatory variables:

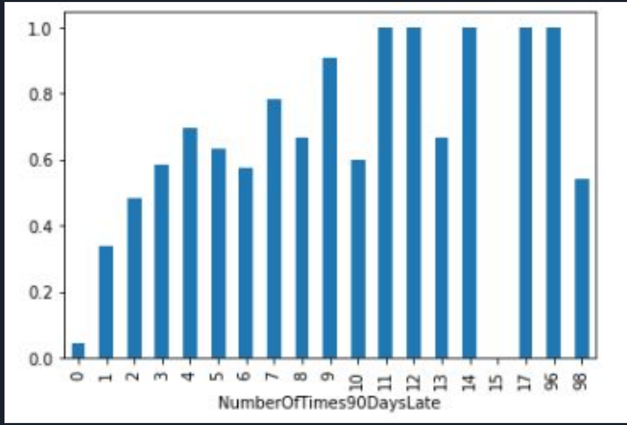
Summary

Variable	Trend	Range of Delinquency Rate	Rank
'NumberOfTimes90DaysLate'	Increases with number of days	2% - 100%	1
'NumberOfTime60-89DaysPastDueNotWorse'	Increases with number of days	2% - 100%	2
NumberOfTime30-59DaysPastDueNotWorse'	Increases with number of days	2% - 100%	3
'RevolvingUtilizationOfUnsecuredLines'	Increases with increase in deciles	2.5% - 25%	4
'NumberRealEstateLoansOrLines'	Increases with number of loans	6%-30%	5
NumberOfOpenCreditLinesAndLoans'	Highest for 0	25% - 6%	6
'age'	Decrease with decrease in deciles	12% - 3%	7
MonthlyIncome'	Decrease with decrease in deciles	4%-8%	8
'DebtRatio'	No strong trend in particular	~ 6%	9
NumberOfDependents'	Remains steady	~ 6%	10

Exploratory Data Analysis: Cleaning







- 96,98 labeled values drop
- Income: replace null with Median
- Number of Dependents : replace null with 0 median
- No significant correlation between features

Metric of consideration:

- Chose Class 1 recall as the metric to optimize:
  - Aim to catch as many delinquencies as possible
  - A high recall for class 1 would mean low False Negatives for Delinquent accounts.
  - Drawback is that it could lead to high False positives, which would mean additional work.

Engineered Features:

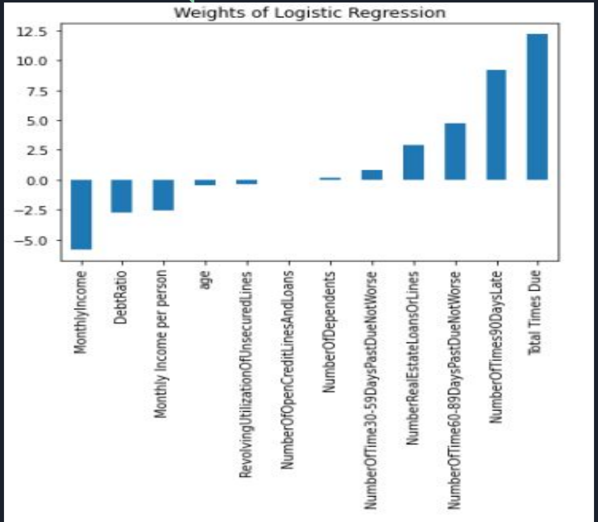
- Total Times Due = N 30-60 + N 60-90 + N90
- Monthly Income/ Person

Baseline: XGBoost with no oversampling

	precision	recall	f1-score	support
0	0.94	0.99	0.97	30922
1	0.56	0.17	0.27	2184
accuracy			0.94	33106
macro avg	0.75	0.58	0.62	33106
weighted avg	0.92	0.94	0.92	33106

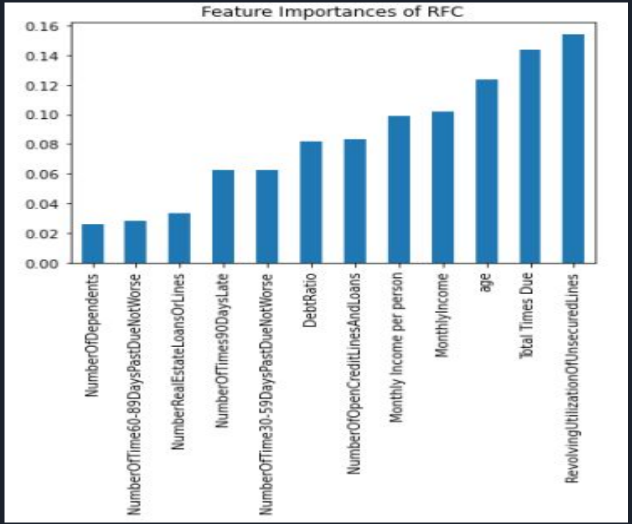
Modeling - Oversampling:

Model 1: Logistic Regression with Oversampling



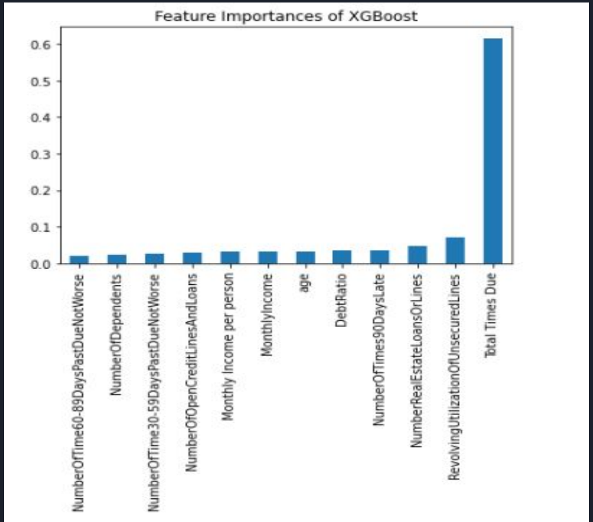
	precision	recall	f1-score	support
0	0.97	0.88	0.92	30922
1	0.26	0.62	0.37	2184
accuracy			0.86	33106
macro avg	0.62	0.75	0.64	33106
weighted avg	0.92	0.86	0.88	33106

Model 2: Random Forest with Oversampling



	precision	recall	f1-score	support
0	0.95	0.98	0.96	30922
1	0.45	0.24	0.32	2184
accuracy			0.93	33106
macro avg	0.70	0.61	0.64	33106
weighted avg	0.92	0.93	0.92	33106

Model 3: XGBoost with Oversampling

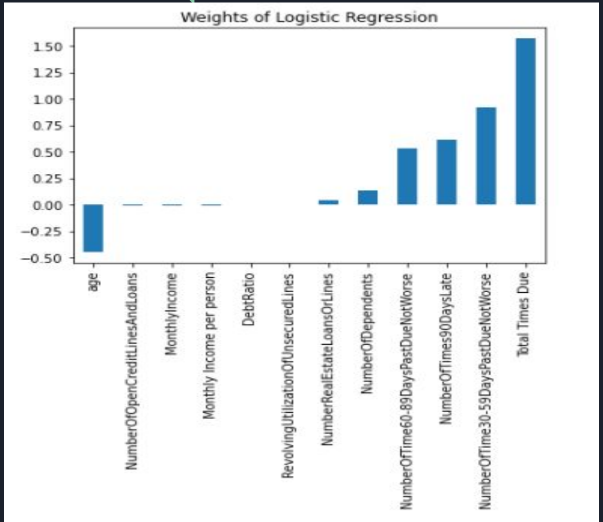


	precision	recall	f1-score	support
0	0.97	0.85	0.91	30922
1	0.24	0.66	0.36	2184
accuracy			0.84	33106
macro avg	0.61	0.76	0.63	33106
weighted avg	0.92	0.84	0.87	33106

Modeling - Oversampling and Tuning:

Model 1: Logistic Regression with Oversampling

'C': 0.000774263682681127, penalty': 'l2'

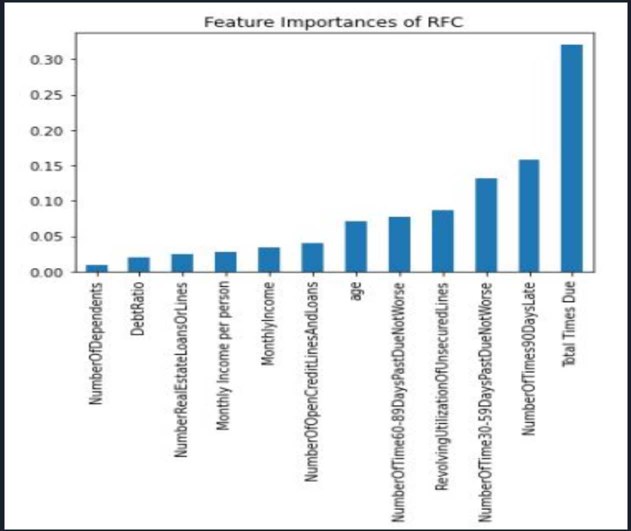


	precision	recall	f1-score	support
0	0.97	0.66	0.79	30922
1	0.12	0.66	0.21	2184
accuracy			0.66	33106
macro avg	0.54	0.66	0.50	33106
weighted avg	0.91	0.66	0.75	33106

Model 2: Random Forest with Oversampling

n_estimators': 50,min_samples_split': 10

10,max_depth': 10,min_samples_leaf': 4

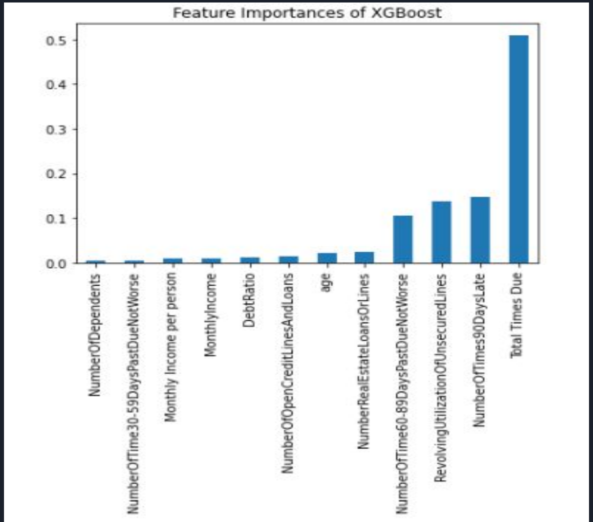


	precision	recall	f1-score	support
0	0.97	0.87	0.92	30922
1	0.26	0.65	0.37	2184
accuracy			0.86	33106
macro avg	0.62	0.76	0.65	33106
weighted avg	0.93	0.86	0.88	33106

Model 3: XGBoost with Oversampling

subsample: 0.8,max_depth': 2,gamma': 0

colsample_bytree': 0.8



	precision	recall	f1-score	support
0	0.98	0.80	0.88	30922
1	0.21	0.76	0.33	2184
accuracy			0.80	33106
macro avg	0.60	0.78	0.61	33106
weighted avg	0.93	0.80	0.84	33106

Modeling - Stack LR, RFC Vs XGBoost:

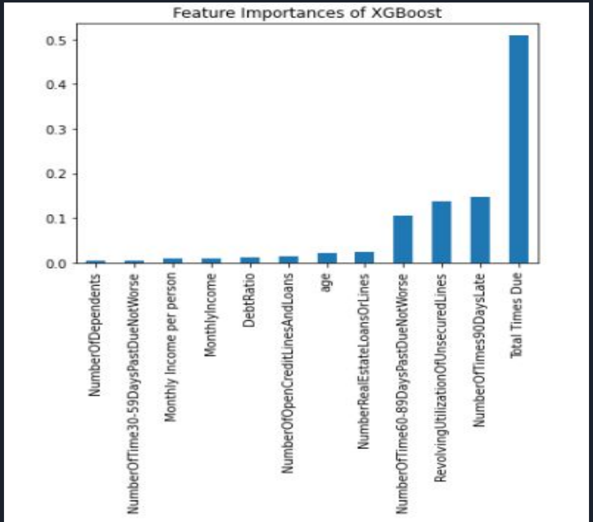
Model 1: Stacked LR, RFC

	precision	recall	f1-score	support
0	0.97	0.85	0.91	30922
1	0.25	0.68	0.36	2184
accuracy			0.84	33106
macro avg	0.61	0.77	0.64	33106
weighted avg	0.93	0.84	0.87	33106

Model 2: XGBoost with Oversampling

subsample: 0.8,max_depth': 2,gamma': 0

colsample_bytree': 0.8

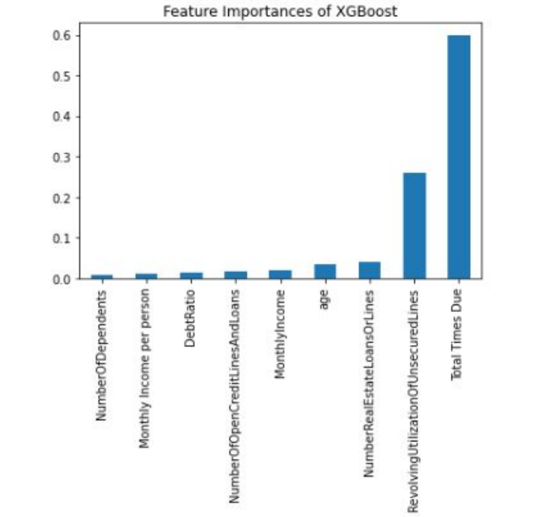


	precision	recall	f1-score	support
0	0.98	0.80	0.88	30922
1	0.21	0.76	0.33	2184
accuracy			0.80	33106
macro avg	0.60	0.78	0.61	33106
weighted avg	0.93	0.80	0.84	33106

Predictions with XGBoost:

Final: XGBoost with Oversampling subsample: 0.8,max_depth': 2,gamma': 0

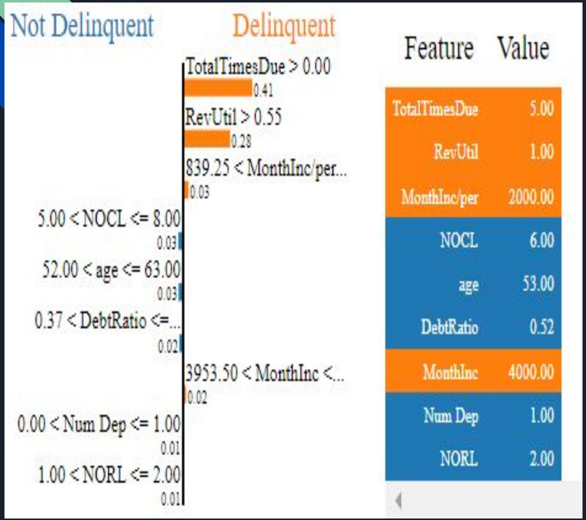
	precision	recall	f1-score	support
0	0.98	0.79	0.87	46151
1	0.21	0.78	0.33	3260
accuracy			0.79	49411
macro avg	0.59	0.78	0.60	49411
weighted avg	0.93	0.79	0.84	49411

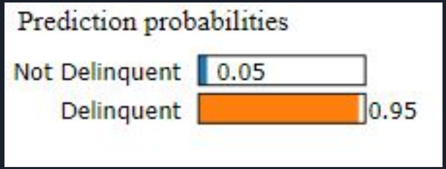


Interpretation:

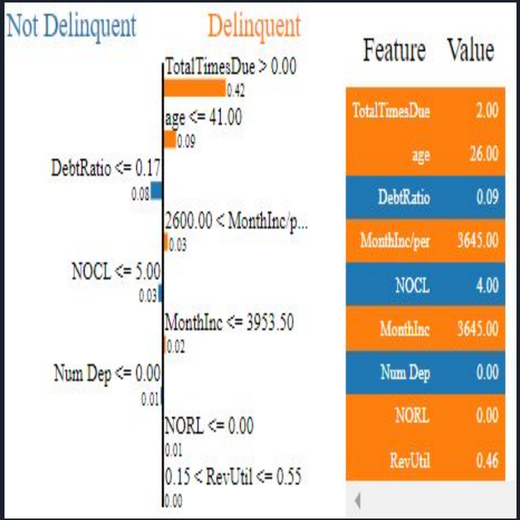
- The recall score for Class 1 is 78%. It means out of all the delinquent accounts, model catches 78% of them.
- The downside is out of all the predictions the model says is delinquent, 21% are actually delinquent.
- The model flags about 25% of the records as delinquent, analysing only these will help weed out 78% delinquent accounts.

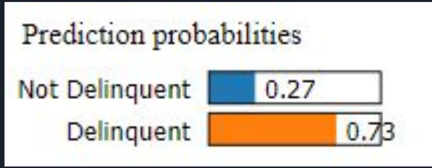
Local Interpretability using LIME:



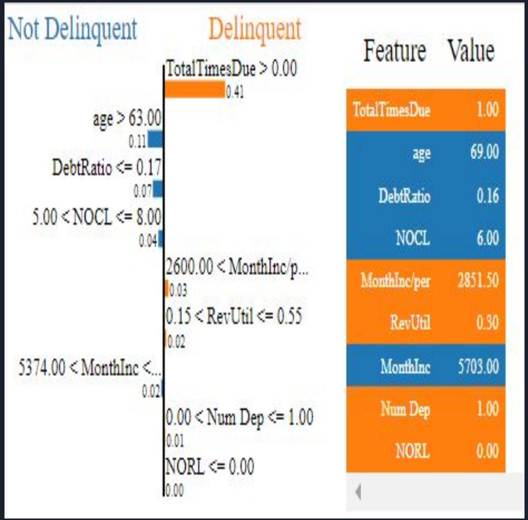


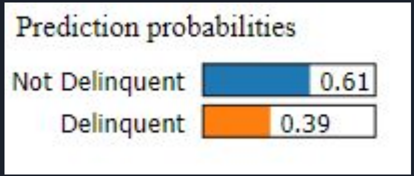
Correctly Classified: Delinquent predicted delinquent





Wrongly Classified: Non Delinquent predicted delinquent





Wrongly Classified: Delinquent predicted Non delinquent

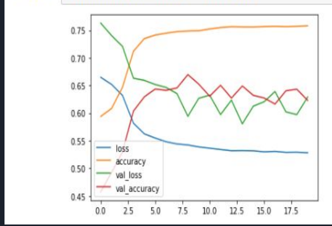
Deep Learning: Used all features including the feature engineered ones with oversampler.

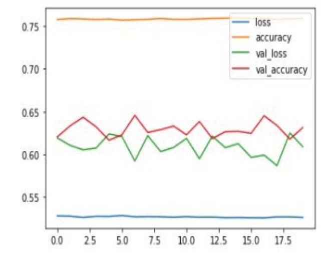
1) 8 nodes in the first layer and 8 in the   second

```
model = Sequential()
model.add(Dense(8, activation ='relu', input_shape =(n_inputs, )))
model.add(Dropout(0.25))
model.add(Dense(8,activation = 'relu'))
model.add(Dropout(0.25))
model.add(Dense(1,activation ='sigmoid'))


model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
In [335]: pd.DataFrame(history.history).plot();
```



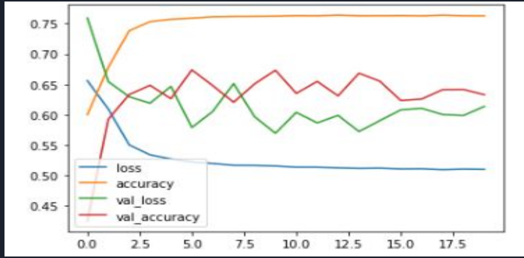


	precision	recall	f1-score	support
0	0.97	0.80	0.88	46151
1	0.20	0.70	0.31	3260
accuracy			0.79	49411
macro avg	0.59	0.75	0.60	49411
weighted avg	0.92	0.79	0.84	49411

2) 16 nodes in the first layer and 16 in the second with

```
n_inputs = X_train_r.shape[1]
model = Sequential()
model.add(Dense(16, activation ='relu', input_shape =(n_inputs, )))
model.add(Dropout(0.25))
model.add(Dense(16,activation = 'relu'))
model.add(Dropout(0.25))
model.add(Dense(1,activation ='sigmoid'))


model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
history = model.fit(X_train_r, y_train_r, epochs=20, validation_split=0.1,batch_size=256)
```



	precision	recall	f1-score	support
0	0.97	0.79	0.87	46151
1	0.19	0.71	0.30	3260
accuracy			0.78	49411
macro avg	0.58	0.75	0.59	49411
weighted avg	0.92	0.78	0.83	49411