Factoring Data Analysis

At the beginning of the problem, I imported IBM Factoring Data using Kaggle API referring to documentation (https://github.com/Kaggle/kaggle-api). After loading the data, saved it as Factoring Data. The full dataset has 2466 entries and no columns have null values.

Below is the description of attributes and their meaning:

IBM Data Attributes	Units	Variable Type	Meaning
countryCode	None	Categorical	Code of the country the customer represents. There are 5 different country code in this dataset.
customerID	None	Categorical	Unique ID assigned to the customer. There are 100 different customers in this dataset.
PaperlessDate	MM/DD/YYYY	Date	Date at which order received
invoiceNumber	None	Numeric	Unique number generated for each order.
InvoiceDate	MM/DD/YYYY	Date	Date at which the invoice is issued, generally the same day as good is sold to the buyer.
DueDate	MM/DD/YYYY	Date	Last date at which the payment is due to the customer for the goods bought.
InvoiceAmount	$ (mostly thousands)	Numeric	Amount for the purchase of goods made by the customer.
Disputed	None	Categorical	Whether there is any dispute of any kind raised by the customer. (ex. Not receiving order, amount)
SettledDate	MM/DD/YYYY	Date	Date at which customer completed the payment for the good delivered.
PaperlessBill	None	Categorical	Type of bill generated. (Electronic/Paper)
DaysToSettle	No. of days	Numeric	No. of days taken by the customer to settle the payment after invoice is raised.
DaysLate	No. of days	Numeric	No. of days the customer is late in making the payment after the due date.

The goal of this project is to predict whether there will be a late payment or not. The task at hand is a supervised, classification and batch learning problem as we have labeled training examples.

The original IBM Factoring Data consisted of 2466 rows and 12 columns representing different attributes. There were no null values in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2466 entries, 0 to 2465
Data columns (total 12 columns):
countryCode      2466 non-null int64
customerID       2466 non-null object
PaperlessDate    2466 non-null object
invoiceNumber    2466 non-null int64
InvoiceDate      2466 non-null object
DueDate          2466 non-null object
InvoiceAmount    2466 non-null float64
Disputed         2466 non-null object
SettledDate      2466 non-null object
PaperlessBill    2466 non-null object
DaysToSettle     2466 non-null int64
DaysLate         2466 non-null int64
dtypes: float64(1), int64(4), object(7)
memory usage: 231.3+ KB
```

Looking at the dataset, we come to know that countryCode and customerID are categorical variables having 5 and 100 different categories respectively.

```
391    616
406    561
770    506
897    396
818    387
Name: countryCode, dtype: int64
```

The we created a new column i.e. Late which checks if the customer was late in making the payment and assigns 1 for late payment and 0 for timely payment. This is the label column for our dataset. After that we converted all date columns (PaperlessDate, InvoiceDate, SettledDate, DueDate) to datetime format in pandas.

In this dataset, I split the training and test set not based on randomly or stratified sampling methods. As in this problem, we would have to predict that in the future whether there will be a late payment or not. So, I split training and test set based on dates. Taking prior dates (in this problem 08/01/2013) as training and dates after prior as test. Attributes like

SettledDate, DaysToSettle, and DaysLate will not be available in the test set as we must predict if payment is going to be made or not, these attributes clearly tell us that.

I then added two more attributes one is the probability of customer late payment and probability to which the customer belongs to its late payment. They are calculated as:

Prob. of customer late payment = $\frac{Sum\ of\ total\ no.of\ times\ late\ payment\ made\ by\ customer}{Sum\ of\ total\ orders\ placed\ by\ the\ customer}$

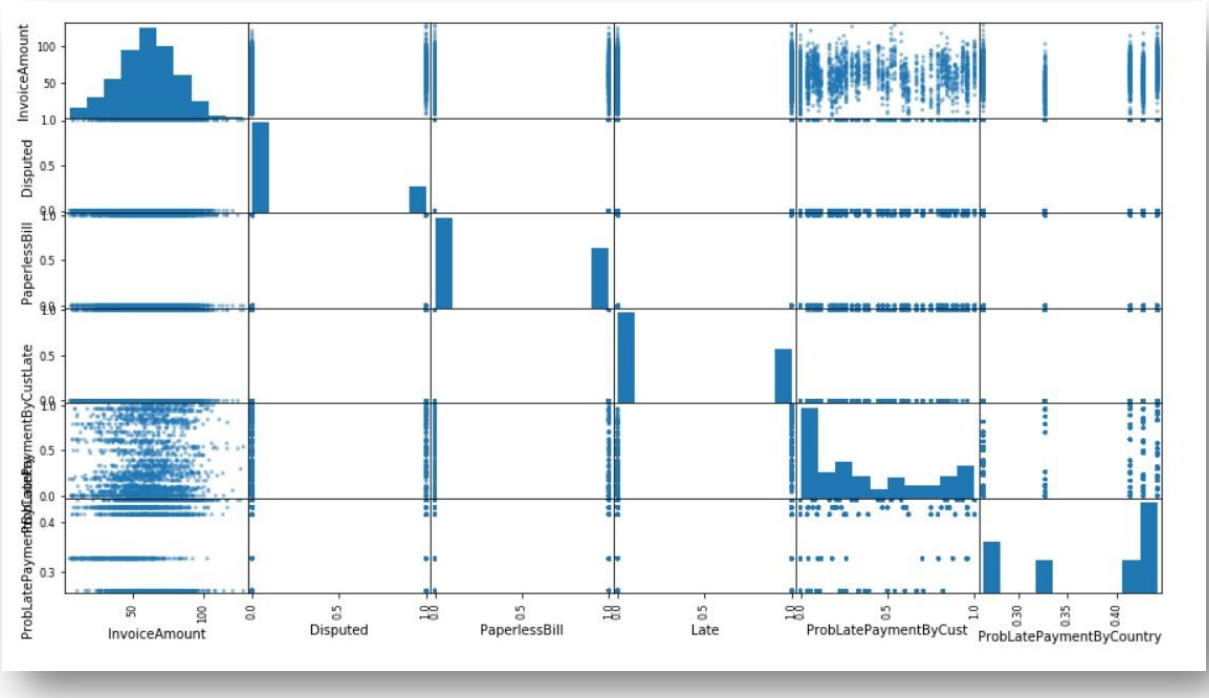
Prob. of country late payment = $\frac{Sum\ of\ total\ no.of\ times\ late\ payment\ made\ by\ all\ customer\ in\ a\ given\ country}{Sum\ of\ total\ orders\ placed\ by\ the\ all\ customer\ in\ a\ given\ country}$

In order to simplify the attributes Disputed and Paperless. I mapped NO:0 and YES:1 for Disputed, and Paper:0 and Electronic:1 for Paperless bill so that they can be analyzed in the algorithm.

Correlation Matrix:

```
Late                          1.000000
ProbLatePaymentByCust         0.702452
Disputed                      0.377497
ProbLatePaymentByCountry      0.149853
InvoiceAmount                 0.065159
PaperlessBill                -0.144638
Name: Late, dtype: float64
```

This shows a high correlation between ProbLatePaymentByCust and Disputed. We also drop attributes InvoiceAmount and PaperlessBill before feeding into the model, as we can see they have less correlation.



Part One: Original Data + Macro-variables:

Average Cross-Validation Scores					
	Logistic	SVM	Random Forest	Voting	Boosted Forest
Accuracy	0.86	0.86	0.86	0.86	0.75
Precision	0.83	0.81	0.81	0.82	0.66
Recall	0.79	0.81	0.81	0.8	0.7

I have used average of cross-validation scores on the train data to come up with the above evaluation metrics. In the code, I have also set max_depth=3 for random forest classifier, boosted forest classifier, and setting C=1 in LinearSVC to avoid overfitting.

For classifying late payment or not, we would prefer a classifier that has good accuracy and high recall and good precision.

I used Random Forrest Classifier as it has high recall compared to the rest, and a good amount of accuracy as well.

Part Two: Original Data + Macro-variables + Financial Variables:

Average Cross-Validation Scores					
	Logistic	SVM	Random Forest	Voting	Boosted Forest
Accuracy	0.86	0.86	0.84	0.86	0.75
Precision	0.82	0.81	0.81	0.82	0.67
Recall	0.79	0.8	0.79	0.79	0.68

I have used average of cross-validation scores on the train data to come up with the above evaluation metrics. In the code, I have also set max_depth=3 for random forest classifier, boosted forest classifier, and setting C=1 in LinearSVC to avoid overfitting.

For classifying late payment or not, we would prefer a classifier that has good accuracy and high recall and good precision.

I used Support Vector Classifier as it has high recall compared to the rest, and a good amount of accuracy as well.

I first downloaded all the data for macroeconomic and financial variables from the given links for dates between 11-01-2011 to 01-01-2014.

For Macro-variables I took the percentage-change to previous time when the data was reported, to see if there was a change in the economic landscape that might have affected payment the customer makes.

For Financial-variables I took the percentage-change to 30 days prior, to see if there was a change that might have affected the markets resulting to delay in payment by customers.

Then I took invoice date and joined the macroeconomic and financial data with respect to each other.

```
Late                          1.000000
ProbLatePaymentByCust         0.702452
Disputed                      0.377497
ProbLatePaymentByCountry      0.149853
InvoiceAmount                 0.065159
DTB3                          0.055432
INDPRO                        0.030707
WILL5000INDFC                 0.017799
CS                            0.015947
PPIACO                       -0.001318
VIXCLS                       -0.014283
ICSA                         -0.018722
PaperlessBill                -0.144638
Name: Late, dtype: float64
```

IBM data: This data is important in finding out if the customer had a history of late payment, if the country the customer belonged to had a history of late payment and whether any dispute was raised on the invoice.

Macro-economic data: We can see in the above image that Macroeconomic production is relatively more important than Inflation rate and Unemployment.

Financial data: Adding financial data does not make the models more effective. I think, this is because financial and macroeconomic data tell us the health of the economy, having one is enough.

I recommend the random forest classifier model and would only use the macro-variables data. Results on test set are displayed below:

```
from sklearn.metrics import precision_score, recall_score, accuracy_score
precision_score(result, y)
```

```
0.7563025210084033
```

```
recall_score(result, y)
```

```
0.656934306569343
```

```
accuracy_score(result, y)
```

```
0.8228438228438228
```

This model also works the best on the test set compared to other models.

REPORT

Goal: Trying to predict whether there will be a late payment on the invoice or not.

Data: IBM Factoring Data, Macroeconomic variables, Financial variables

The factoring data is important to find patterns on customer behavior, how people from different country behave in relation to making payments and whether there was any dispute raised. We were able to find significant correlation with late payment if the customer had the history of the same and if it was a dispute was raised.

The macroeconomic variables that we used were unemployment (ICSA), production (INDPRO) and inflation rate (PPIACO) to see if there were any trends related to late payment due to change in macroeconomic conditions. These were little useful in prediction.

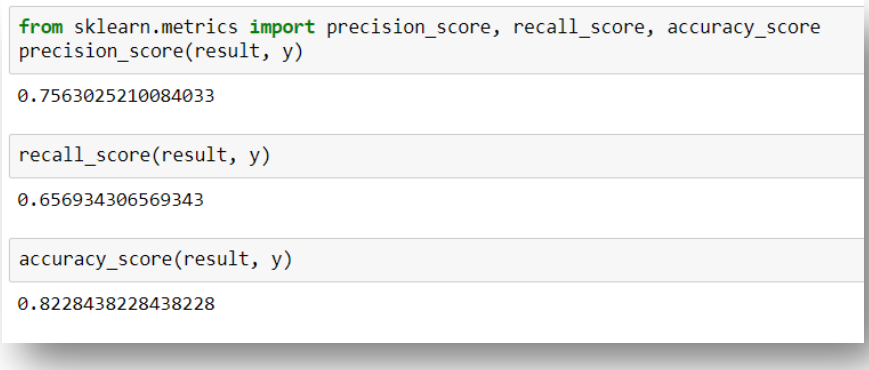
The financial variables that we used were stock market returns (WILL5000INDFC), stock market volatility (VIXCLS), risk-free rate (DTB3), and credit spread between Baa and Aaa corporate bonds (CS). These variables did not add any significant value as all the information regarding the economic conditions that could result in a delay of payment was already contained in the macroeconomic variables.

Correlation with respect to late payment:

```
Late                          1.000000
ProbLatePaymentByCust         0.702452
Disputed                      0.377497
ProbLatePaymentByCountry      0.149853
InvoiceAmount                 0.065159
DTB3                          0.055432
INDPRO                        0.030707
WILL5000INDFC                 0.017799
CS                            0.015947
PPIACO                       -0.001318
VIXCLS                       -0.014283
ICSA                         -0.018722
PaperlessBill                -0.144638
Name: Late, dtype: float64
```

Model: In order to solve this classification problem, we used various Machine Learning models like logistic regression, support vector machine, random forest classifier, ensemble voting classifier, and boosted forest methods.

Result: The best model that we selected from our analysis was the random forest classifier as its performance was best relative to the others, we also just used macroeconomic variables as adding financial variables does not improve the model efficiency.



The above are the results we got for the model on our test data (data set aside at the beginning of the project to check how would it behave in the real world). The accuracy of the result is 82%, precision is 76% and recall is 66%.

To give context to these numbers, if we did not use any model and predicted that there would be no late payments. The accuracy would still be 64%.

In this problem we were also focused on having a high recall comparative to precision as we did not want many false negatives as we would take those payments and it would result in being late.

The model is 18% more accurate compared to the assumption of all loans being on-time payments.

Recommendation: This model is not phenomenal in classifying factoring data for late payment but is a fairly good. It also takes into consideration factors that would be difficult to quantify subjectively. This could be our first filter in to choosing whether we would like to take the receivables or not. It would also reduce time and effort of analyzing all receivables in the beginning individual on which a lot of man-hours are lost.