Final Project

Abir Razzak  
INFO371 – Data Mining Applications  
December 10, 2019

Table of Contents

[Getting Started 3](#_Toc26726673)

[Error Handling (ARFF Formatting) 3](#_Toc26726674)

[Initial Data Analysis 4](#_Toc26726675)

[Exploratory Data Analysis 5](#_Toc26726676)

[Demographic Variables 5](#_Toc26726677)

[SEX 5](#_Toc26726678)

[EDUCATION 6](#_Toc26726679)

[MARRIAGE 6](#_Toc26726680)

[AGE 7](#_Toc26726681)

[Payment Status Variables 7](#_Toc26726682)

[PAY\_0 8](#_Toc26726683)

[PAY\_2 9](#_Toc26726684)

[PAY\_3 10](#_Toc26726685)

[PAY\_4 11](#_Toc26726686)

[PAY\_5 11](#_Toc26726687)

[PAY\_6 12](#_Toc26726688)

[Data Classification 13](#_Toc26726689)

[Naïve Bayes Classification 14](#_Toc26726690)

[Results: 14](#_Toc26726691)

[Decision Tree 15](#_Toc26726692)

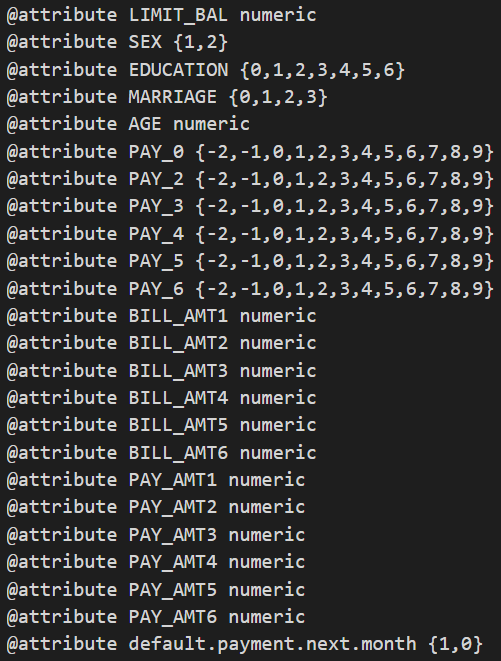
[Results: 15](#_Toc26726693)

[Discussion 16](#_Toc26726694)

# Getting Started

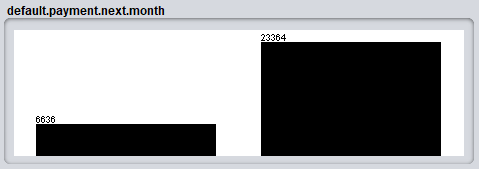
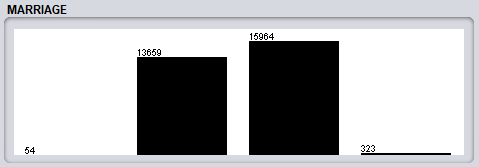
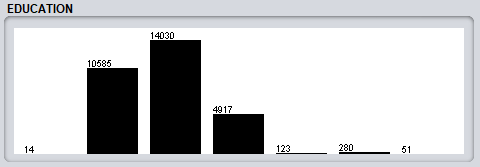
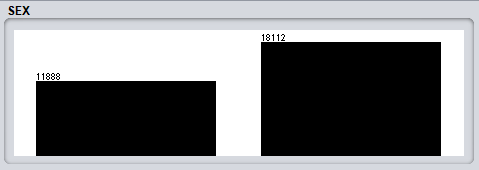
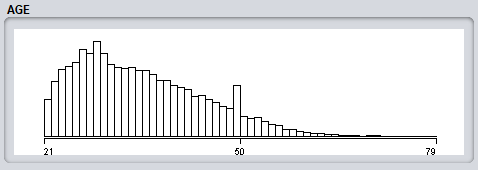
First thing we have to do is to save the .csv file into a proper .arff file. In Weka, I opened up the csv file and then clicked “save as” and saved the new file as a .arff file. However, there were some changes that needed to be made to the .arff file, as all the attributes were tagged as numeric data instead of categorical, such as Sex, Marriage, and Education. Using the help of the description box on Kaggle, I was able to figure out which attributes were categorical or not and helped me figure out what the categories meant. Below is a list of the attributes and their type/categories.

## Error Handling (ARFF Formatting)

Using these as the attributes, I got an error message on Weka telling me that my first line of data was incorrect. Upon further inspection, it looks like PAY\_X attributes can have a value of -2 even though the Kaggle page does not specify this. After fixing that, the next line that was throwing an error had a value of 0 in the PAY\_X attributes, which again were not specified in the Kaggle page. After adding both -2, and 0 to the list of possible values for PAY\_X, again I got another error. This time the error had to do with the marriage attribute. There was a value of 0 for MARRIAGE, the Kaggle description doesn’t mention what a value of 0 is supposed to mean, but after examining the data, there were 54 instances of MARRIAGE being a value of 0, so I added 0 to the list of categories for that. Next error I got were that there were 14 instances where EDUCATION equals 0. Again, Kaggle doesn’t mention anything about EDUCATION being equal to 0, but I added a category for it anyways since there were more than 1 instance causing a problem. And finally with all that out of the way, the .arff file was finally accepted by Weka with no more error messages popping up anymore. Once the .arff file opened, I realized we do not need the ID attribute in the dataset, so removed that attribute from all the instances from within Weka and saved the new .arff file. After all the changes, the attributes look like the following:

## Initial Data Analysis

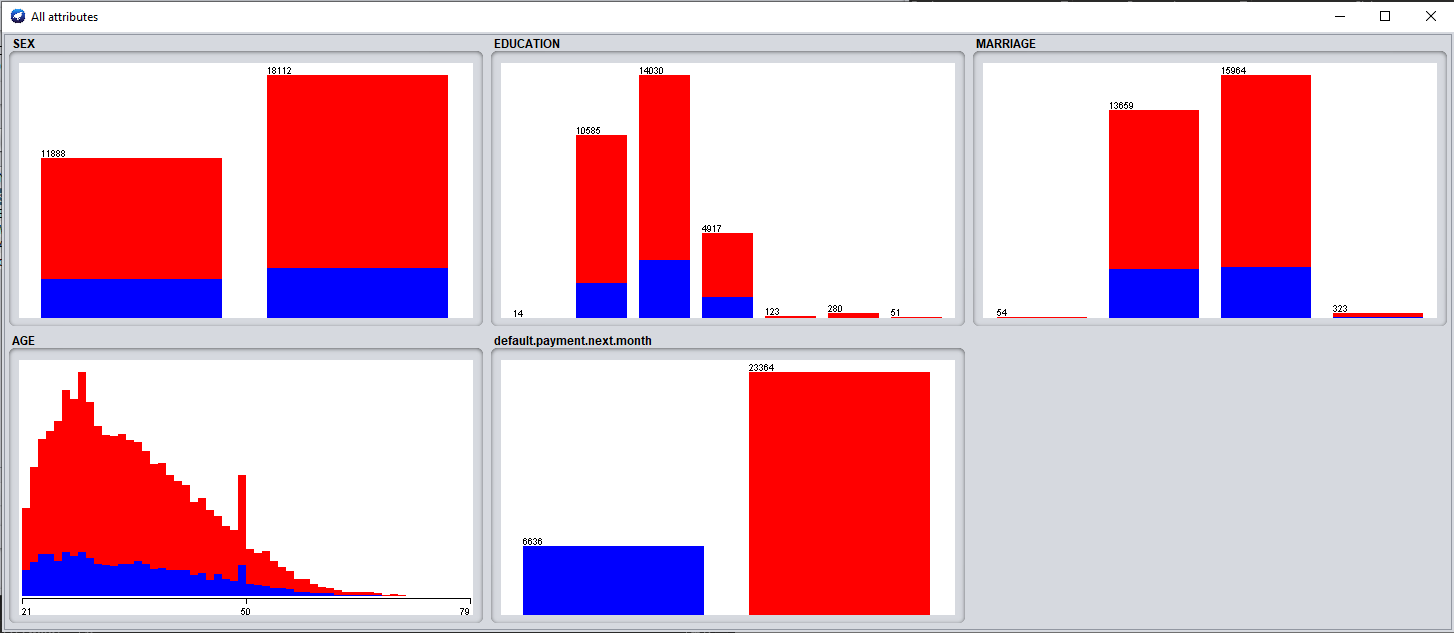
Here are some screenshots of the raw data inside Weka



# Exploratory Data Analysis

Here I will explain the different variables and how they might have an impact on whether or not that instance has a default payment next month. In all visualizations in this section, the color blue represents “1”, meaning Yes, they will have a default payment, and the color red represents “0”, meaning No, they will not have a default payment.

## Demographic Variables

The demographic variables in the dataset are the attributes: SEX, EDUCATION, MARRIAGE, and AGE. When looking at the spreads its important to take note that overall, the number of No defaults outweigh the number of Yes defaults by quite a margin. Because of this, for my analysis I will conduct comparisons in matter of percentages rather than counts of instances.  


### SEX

Taking a look at the SEX graph, we can see that roughly the same number of Yes defualts are in each gender. From a proportions perspective, this then means that males are more likely to have a default payment than females. Males, with 2873 instances of default payments, means that roughly 24.17% of males in this dataset have a default on their next payment. Females, with 3763 instances of default payments, means that roughly 20.78% of females in this dataset have a default on their next payment. Therefore, being a male in this dataset puts you at a slightly higher chance of having a default on your next payment.  


### EDUCATION

At a first glance, EDUCATION seems to be mostly concentrated around those instances that attend some form of schooling (highschool and up). After running some calculations, instances with only a high school education had the highest percetage of indivisuals who have a default on their payment. Shortly behind are instances with university education, and then shortly behind that are instances with graduate school education. After those with schooling, unknown #6 was slightly behind graduate school instances, and after that the drop occurred from 15.69% to 6.43%. Surprisingly, having an unknown education with a value of 0 had not a single instance of having a default on their payment, giving them a 0%. Although, we do not know if this is statistically valid, as there are only 14 instances of having a 0 in value for EDUCATION. Therefore, having some kind of formal schooling puts an instance at a higher chance of having a defualt on their payment.  

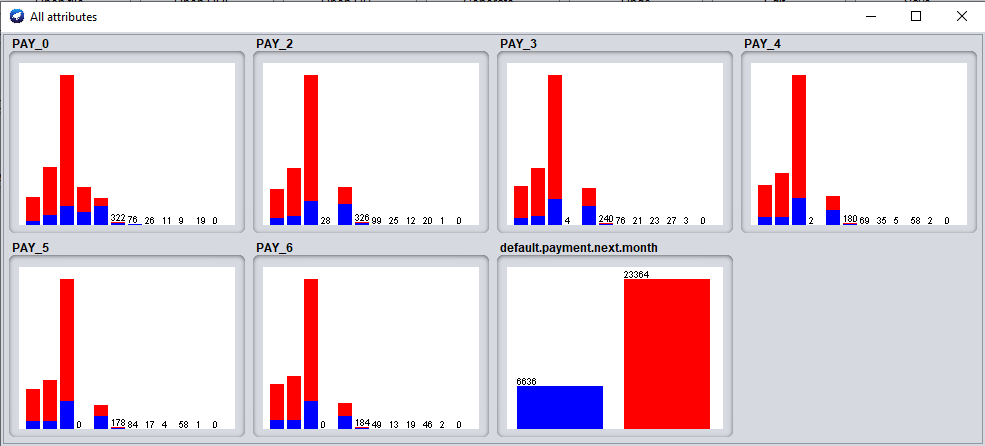

### MARRIAGE

For MARRIAGE there is a pretty even split amongst the different marriage categories. A value of 3 (Others) takes the highest percentage of 26.01%, followed by 1 (Married) with 23.47%, and lastly 2 (Single) with 20.93%. The Kaggle page does not indicate what 0 is supposed to mean, therefore I marked is as unknown; that value had a percentage of 9.26% which is under half of what the other values’ percentages were, however because we don’t know what this value is supposed to mean, I will disregard it in the conclusion. Although the percentage of 3 (Others) is the highest, there are only 323 instances of this in the dataset. Compared to 1 (Married) and 2 (Single) which have counts of over 13,000 instances, the total number of “Others” marriage status is very low. Because of its lower count, I consider this category to be a bit volitile, and as more data enters this dataset, I expect the percentage for “Others” to normalize around 21%-23% like the other two categories. Therefore, marriage status is roughly the same when it comes to determining defaults on their payments, and does not have a significant impact in trying to predict an instance’s default on payment.  


### AGE

Observing the graph for AGE, we can see that both the red and blue tend to follow roughly the same patterns, with a slightly higher peak for No defaults around the age of 28-29 years old. After that peak, we observe a gradual decrease in overall instances as age increases. We observe that both defualts and no defaults are decreasing at about the same rate proportional to their total number; we observe what looks like a linear decrease as age increases for defaults in payment, and a gradual curved decrease as age increases for no defualts in payment. Because of this, the percetage between the two categories look to be about the same across most AGE buckets. In both default payment cases, we observe a strange peak around the age of 49-50. However, the peak still maintains the proportions of defult payment and no defualt payment. After that peak, it looks like their gradual decreases pick back up where they were supposed to. Because both the red and the blue parts of the graph tend to maintain their proportions to one another throughout all ages in the dataset, I conclude that AGE does not have a very significant impact on whether or not there will be a default on their payment.

## Payment Status Variables

The payment status variables in the dataset are the attributes: PAY\_0, PAY\_2, PAY\_3, PAY\_4, PAY\_5, and PAY\_6. When looking at the spreads its important to take note that overall, the number of No defaults outweigh the number of Yes defaults by quite a margin. Because of this, for my analysis I will conduct comparisons in matter of percentages rather than counts of instances. I believe that these attributes will help greatly with prediciting if there will be a default in payment, because these attributes show each instances historical information about how delayed they pay off their credit card payments.  


### PAY\_0

I expect out of all the payment status variables, that PAY\_0 should give a classifier the best information on whether or not the payment will be defualted since this attribute would be the most recent piece of data from all the others. Because the original Kaggle description does not mention what a value of -2 or 0 means in the context of the dataset, I will exclude those rows from my final conclusion. We observe a wide range of percentages in PAY\_0, with high of 75.78% of 3 month delayers defualting their payment, and a low of 16.78% of duly payers defualting their payment. Overall, we observe that the percentages are much higher on 1+ month delayers (value of 1-9) than duly payers (value of -1).This makes sense, as one who pays off their credit card payments without delay are less likely to defualt on a payment, than those who delay their payment by a few months. In conclusion, I believe that the PAY\_0 attribute has an impact on the ability to predict defaulted payments.



### PAY\_2

Looking further back to August, 2005, this data could be considered less valuable than the previous payment status variables, however this data still gives us insight into the historical financial ability for the instance and should be considered a useful piece of information. Because the original Kaggle description does not mention what a value of -2 or 0 means in the context of the dataset, I will exclude those rows from my final conclusion. We observe a wide range of percentages in PAY\_2, with high of 75.00% of 6 month delayers defualting their payment, and a low of 15.97% of duly payers defualting their payment. Overall, we observe that the percentages are much higher on 1+ month delayers (value of 1-9) than duly payers (value of -1). This makes sense, as one who pays off their credit card payments without delay are less likely to defualt on a payment, than those who delay their payment by a few months. In conclusion, I believe that the PAY\_2 attribute has an impact on the ability to predict defaulted payments.



### PAY\_3

Looking further back to July, 2005, this data could be considered less valuable than the previous payment status variables, however this data still gives us insight into the historical financial ability for the instance and should be considered a useful piece of information. Because the original Kaggle description does not mention what a value of -2 or 0 means in the context of the dataset, I will exclude those rows from my final conclusion. We observe a wide range of percentages in PAY\_3, with high of 81.48% of 7 month delayers defualting their payment, and a low of 15.59% of duly payers defualting their payment. Overall, we observe that the percentages are much higher on 1+ month delayers (value of 1-9) than duly payers (value of -1). This makes sense, as one who pays off their credit card payments without delay are less likely to defualt on a payment, than those who delay their payment by a few months. In conclusion, I believe that the PAY\_3 attribute has an impact on the ability to predict defaulted payments.



### PAY\_4

Looking further back to June, 2005, this data could be considered less valuable than the previous payment status variables, however this data still gives us insight into the historical financial ability for the instance and should be considered a useful piece of information. Because the original Kaggle description does not mention what a value of -2 or 0 means in the context of the dataset, I will exclude those rows from my final conclusion. We observe a wide range of percentages in PAY\_4, with high of 82.76% of 7 month delayers defualting their payment, and a low of 15.90% of duly payers defualting their payment. Overall, we observe that the percentages are much higher on 1+ month delayers (value of 1-9) than duly payers (value of -1). This makes sense, as one who pays off their credit card payments without delay are less likely to defualt on a payment, than those who delay their payment by a few months. In conclusion, I believe that the PAY\_4 attribute has an impact on the ability to predict defaulted payments.



### PAY\_5

Looking further back to May, 2005, this data could be considered less valuable than the previous payment status variables, however this data still gives us insight into the historical financial ability for the instance and should be considered a useful piece of information. Because the original Kaggle description does not mention what a value of -2 or 0 means in the context of the dataset, I will exclude those rows from my final conclusion. We observe a wide range of percentages in PAY\_5, with high of 82.76% of 7 month delayers defualting their payment, and a low of 16.19% of duly payers defualting their payment. Overall, we observe that the percentages are much higher on 1+ month delayers (value of 1-9) than duly payers (value of -1). This makes sense, as one who pays off their credit card payments without delay are less likely to defualt on a payment, than those who delay their payment by a few months. In conclusion, I believe that the PAY\_5 attribute has an impact on the ability to predict defaulted payments.



### PAY\_6

Looking further back to April, 2005, this data could be considered less valuable than the previous payment status variables, however this data still gives us insight into the historical financial ability for the instance and should be considered a useful piece of information. Because the original Kaggle description does not mention what a value of -2 or 0 means in the context of the dataset, I will exclude those rows from my final conclusion. We observe a wide range of percentages in PAY\_6, with high of 82.61% of 7 month delayers defualting their payment, and a low of 16.99% of duly payers defualting their payment. Overall, we observe that the percentages are much higher on 1+ month delayers (value of 1-9) than duly payers (value of -1). This makes sense, as one who pays off their credit card payments without delay are less likely to defualt on a payment, than those who delay their payment by a few months. In conclusion, I believe that the PAY\_6 attribute has an impact on the ability to predict defaulted payments

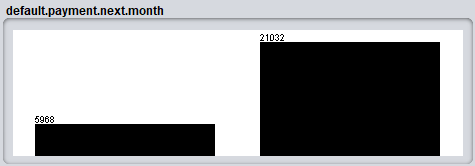
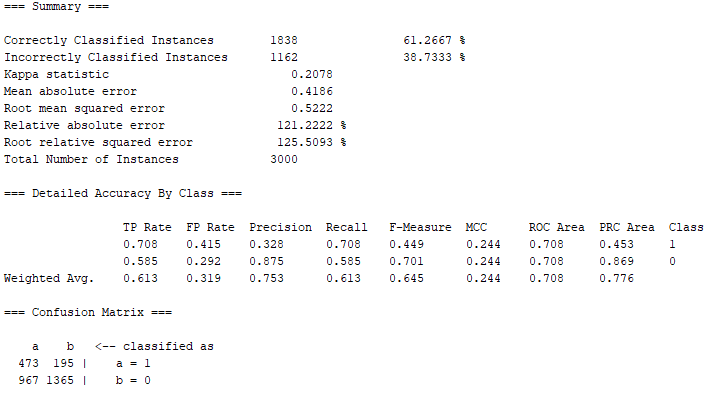


Overall, across the payment status variables, there were some patterns that I noticed. Every attribute’s category resulting in the lowest percentage was -1 (Duly Payer). This makes sense, since one who pays their credit card payments are less likely to default on a payment. The interesting part was that across all the payment status variables, the percentage of duly payers who defaulted were around 15% to 16%, which I found very interesting. Across all the variables, 2 (2 months delayed) was always at least 50%, then in most cases from 2 to 9 (9+ months delayed) that precentage increased as the months delayed increased. This is a direct correlation, and I suspect that the classification models that will be used later on this dataset will utilize these payment status variables to help them accruately predict if an instance defaulted on their payment or not.

# Data Classification

In this section, I will use Weka to perform various predictions based on the built-in classifiers that come within the program. I will used various techniques that we have been taught throughtout the quarter to prepare the datasets and to run the classifications on the data.

## Naïve Bayes Classification

To start the Naïve Bayes Classification, first we need to create a training data set. Because the actual dataset contains 30,000 instances, I made the testing dataset 3,000 instances (so that we have a 10%-90% split between training and testing data), therfore the training dataset contains 27,000 instances. For the sake of simplicity, I took the first 3,000 instances as the training data, and the rest as the testing data. Below is the distribution for default payments within the training data:   
  
After running Naïve Bayes Classifier on the testing data using the training data, the following output was achieved:

### Results:

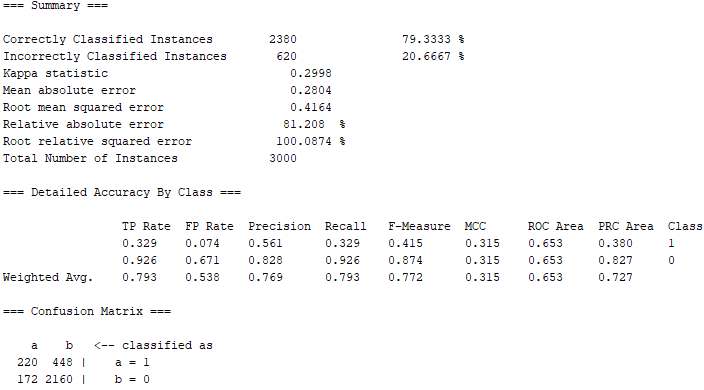
Correctly identified test instances: 1838 / 3000 instances (61.27%)  
Incorrectly identifited test instances: 1162 / 3000 instances (38.73%)

|  |  |  |  |
| --- | --- | --- | --- |
| Default on Payment | *Recall* | *Precision* | *F-Measure* |
| 1 (Yes) | 0.708 | 0.328 | 0.449 |
| 0 (No) | 0.585 | 0.875 | 0.701 |
| **Weighted Average:** | **0.618** | **0.753** | **0.645** |

The classification correctly identified Yes 473 times, and incorrectly identified it as No 195 times. It correctly identified No 1365 times and, incorrectly indentified it as Yes 967 times. It seems as though the classification had a hard time prediciting No, but an easier time prediciting Yes. The classification was slightly skewwed towards prediciting an instance as No (1560 total) than Yes (1440 total). The confusion matrix is pasted below:  


## Decision Tree

Using the same training and testing datasets as the Naïve Bayes Classification, I ran the J48 tree classification with “unpruned” set to false. Below are the results from the classification:



### Results:

The decision tree was far more accurate than the Naïve Bayes classification that was ran in the previous section. Going from a success rate of 61.27% (Naïve Bayes) to 79.33% (Decision Tree), the results were about 29.5% more successful.

Correctly identified test instances: 2380 / 3000 instances (79.33%)  
Incorrectly identifited test instances: 620 / 3000 instances (20.67%)

The classification correctly identified Yes 220 times, and incorrectly identified it as No 448 times. It correctly identified No 2160 times and, incorrectly indentified it as Yes 172 times. The classification had a harder time predicting Yes, as only 220 out of the total 668 instances of actual Yes, were predictied correctly. The classification tended to predict an instance as No (2608 total) more often than Yes (392 total). The confusion matrix is pasted below:  


# Discussion

At the end of the day, the J48 pruned tree classification was the most accurate classification method on my training and testing data. The J48 was about 29.5% more successful at properly classifying instances than the Naïve Bayes. This means that out of the 3,000 instances in the testing data, the decision tree achieved 542 more correctly identified instances than the Naïve Bayes. Looking at the attributes from the second section, the PAY\_X attributes were all significant factors in helping the classification methods predict more easily. With regards to the J48 classification, the method was more successful by enabling pruning. Disabling pruning yielded a success rate of about 74.9%, compared to its pruned counterpart at 79.33%. All in all, at about an 80% success guess rate, I’d say that this method has a pretty good rate of successfully predicting if future instances will have defaulted payments or not.