CIND 119: Final Project

Customer Churn

CIND 119 – D30

Instructor: Zekiye Erdem

Marco Muto

Ifra Zahid

Table of contents

1. Summary …………………………………………………………..………………………………………………….….…………..2
2. Workload Distribution ………………………………………………………………………………………….……………….2
3. Data Preparation …………………………………………………………………………………………………….…………....3
   1. 3.1 Importing the dataset …………………………………………………………………………………………3
   2. 3.2 Summarizing the data …………………………………………………………………………….………..…3
   3. 3.3 Importing the dataset ……………………………………………………………………………….………10
   4. 3.4 Feature Selection …………………………………………………………………………………….……..…12
   5. 3.5 Balancing the Dataset ………………………………………………………………………….………..….14
   6. 3.6 Scaling the Data ………………………………………………………………………………………….....…15
4. Predictive Modelling ……………………………………………………………………………………………………..….…16
   1. 4.1 Baseline Model ………………………………………………………………………………………….………18
   2. 4.2 Classifier Analysis ……………………………………………………………………………………………...21
5. Conclusion ………………………………………………………………………………………………………………….………..22

Customer Churn Prediction

# Members

Marco Muto, marco.muto@ryerson.ca   
Ifra Zahid, izahid@ryerson.ca

1. Summary: Many companies that use a subscriptions/contracts deal with customer churn due to an array of factors. Churn is essentially when a customer decides to move on from a particular product or brand for a competitor. The company that hired us is in the telecom industry and is trying to build a retention strategy to likely improve the ‘know your customer’ aspect of their business. The telecom industry may not have many players in the United states but the rise of smaller companies appealing to clients can be considered a looming threat. If Telco can better predict customer turnover then they can have an easier time learning what factors may help customer retention.

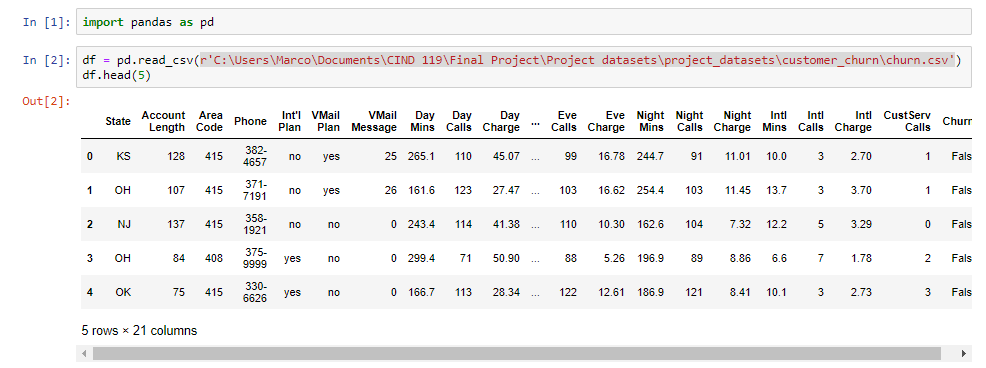
In this paper we will breakdown the data provided by the company. We intend to uncover customers who are likely to churn in the near future. We will do this by explaining how we prepared our data to apply the best classification tool to predict churn. This will help us to potentially reduce the risk of customer churn by providing insights on when this company should take precautionary measures to prevent this from happening.

# Workload Distribution

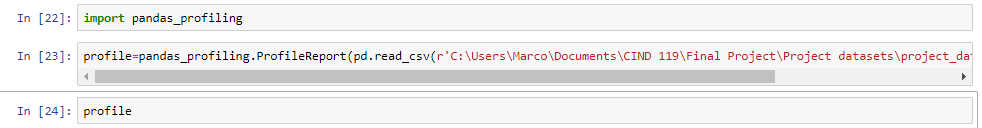
|  |  |
| --- | --- |
| Member Name | List of Tasks Performed |
| Marco Muto | Summary, Data preparation, predictive modeling/Classification and conclusions |
| Ifra Zahid | Predictive Modeling/Classification description |

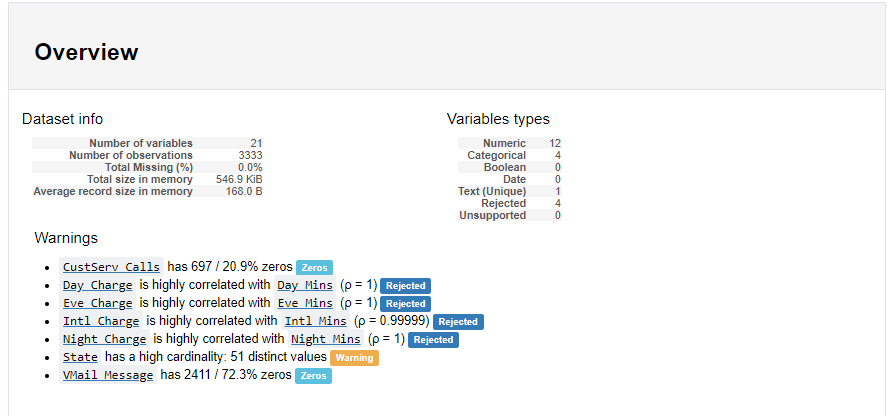
# Data Preparation

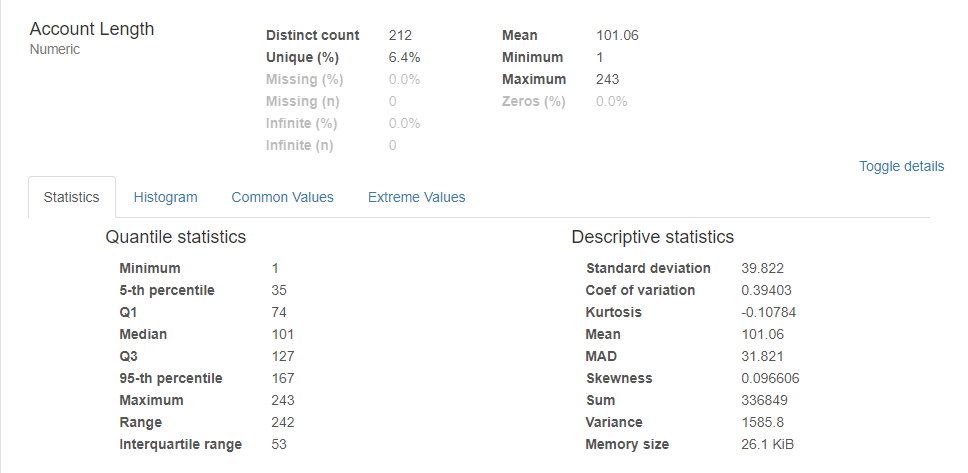
**3.1 Importing the dataset**

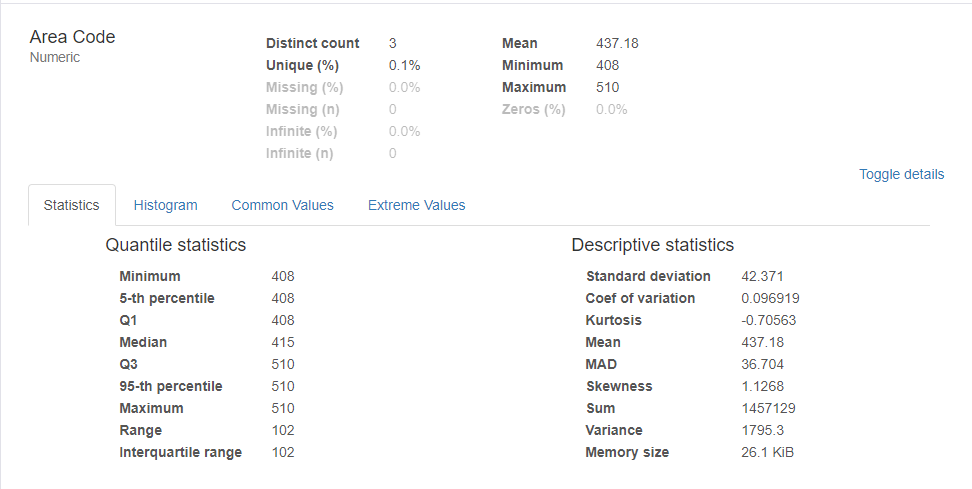
First, we need to start out by importing the comma separated value document with pandas in python. We’ll also execute the df.head() command to check if the data was imported correctly.

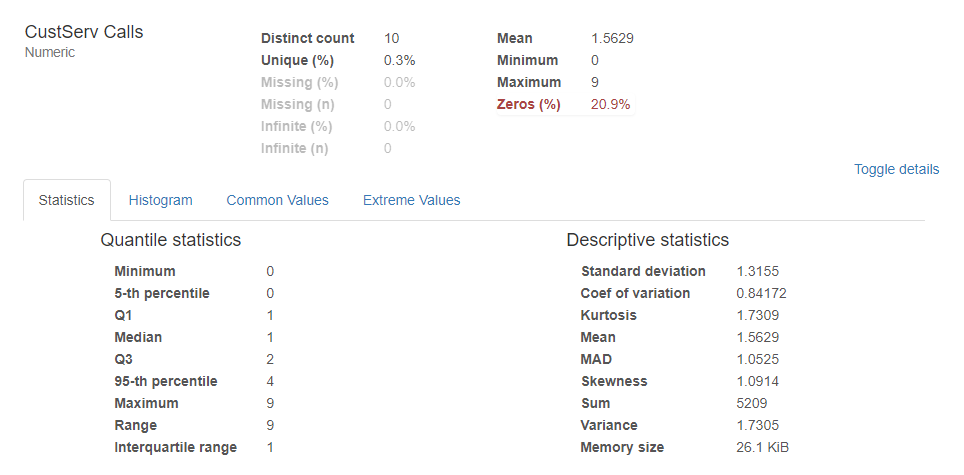
**3.2 Summarizing the data**

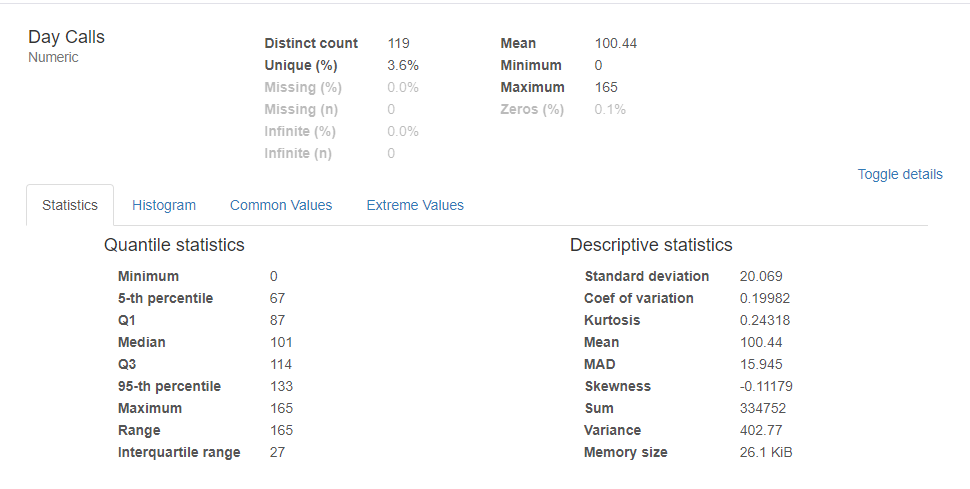
Next, we can easily perform exploratory analysis with pandas profiling’s report function. With just a few lines of code we can generate a report analyzing the data within.

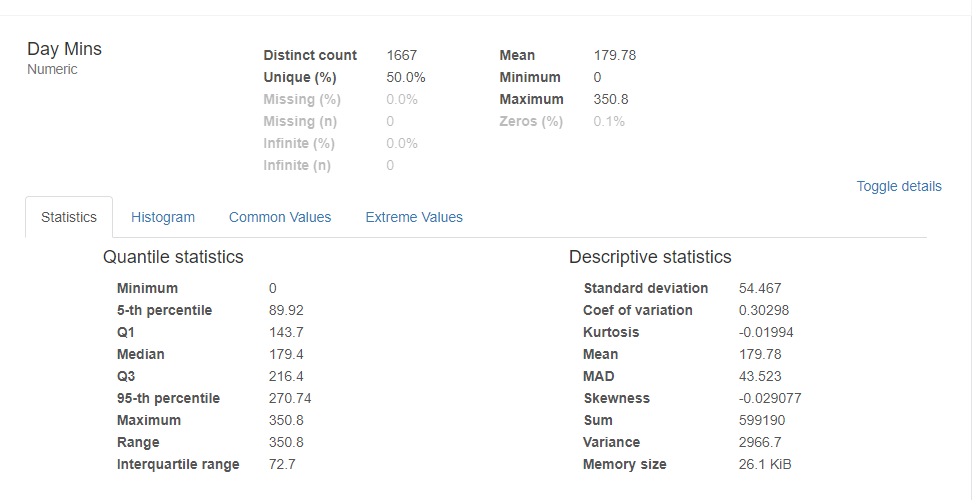


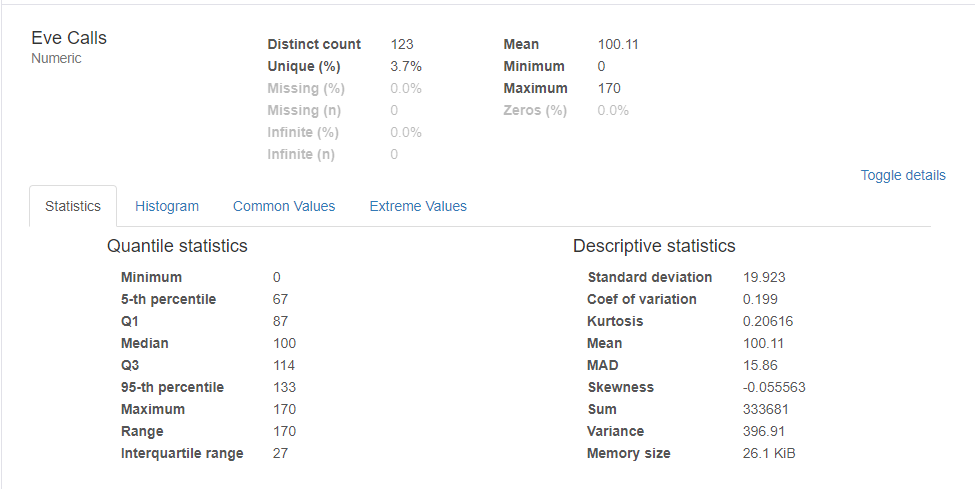




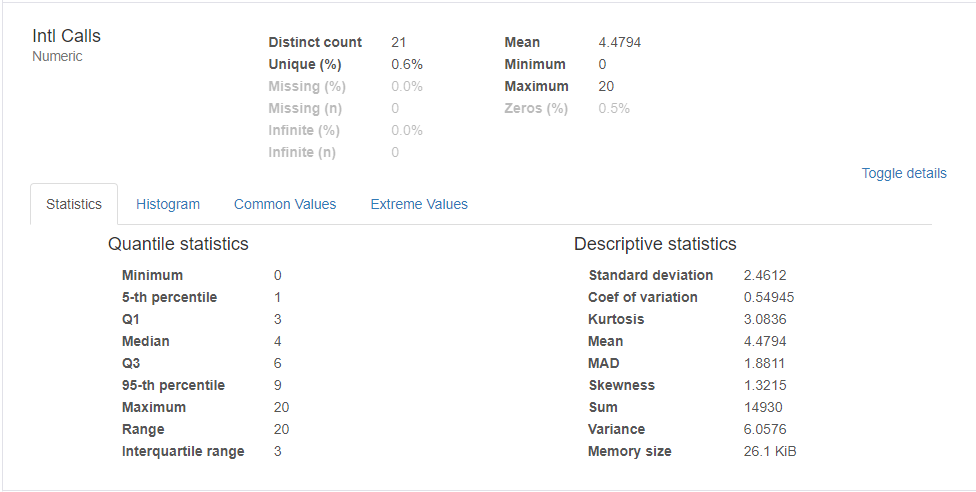


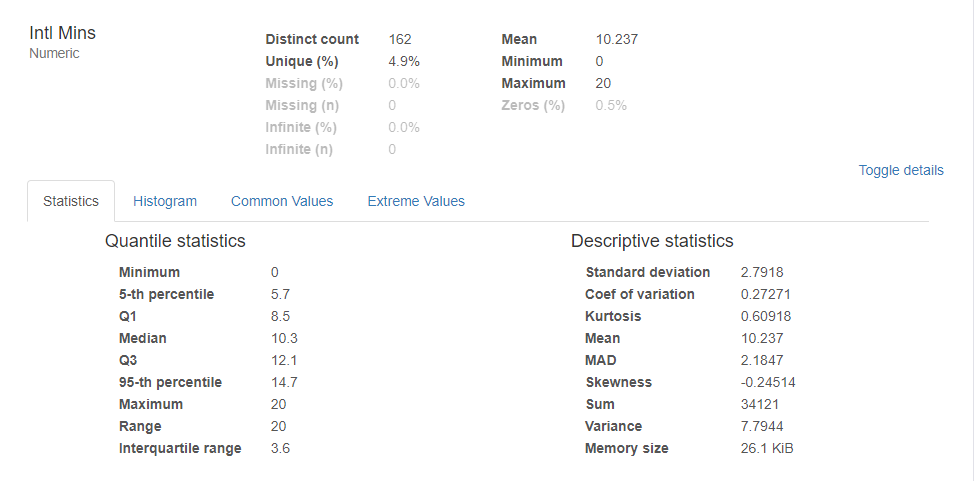


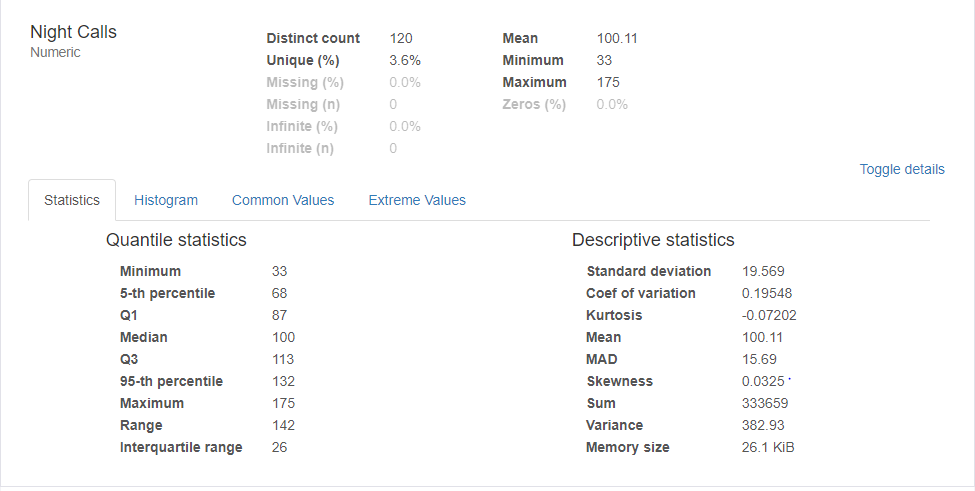


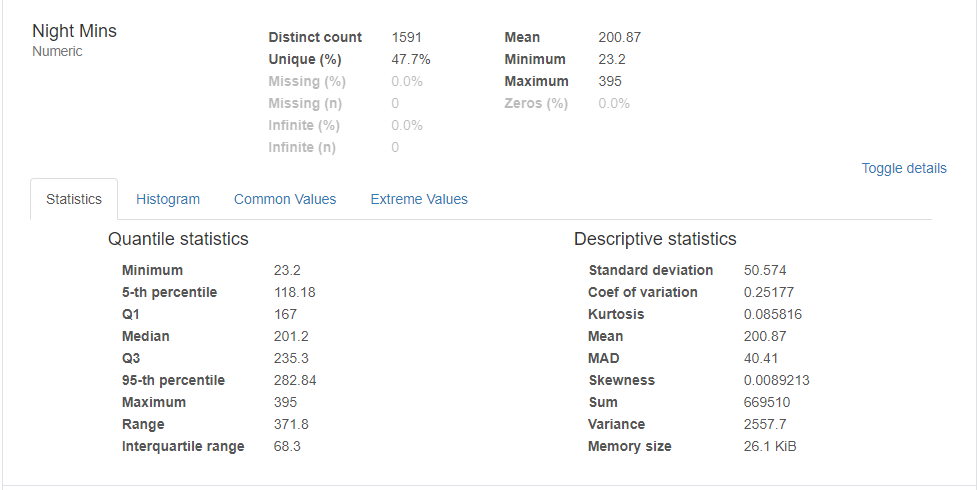


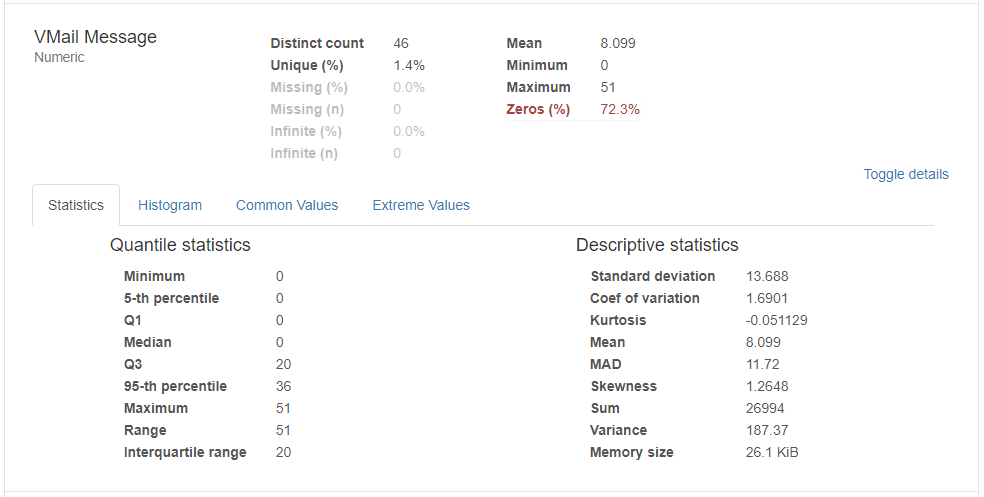




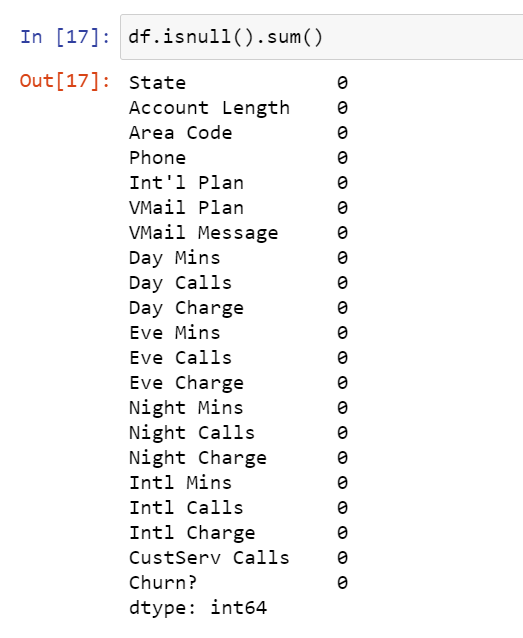




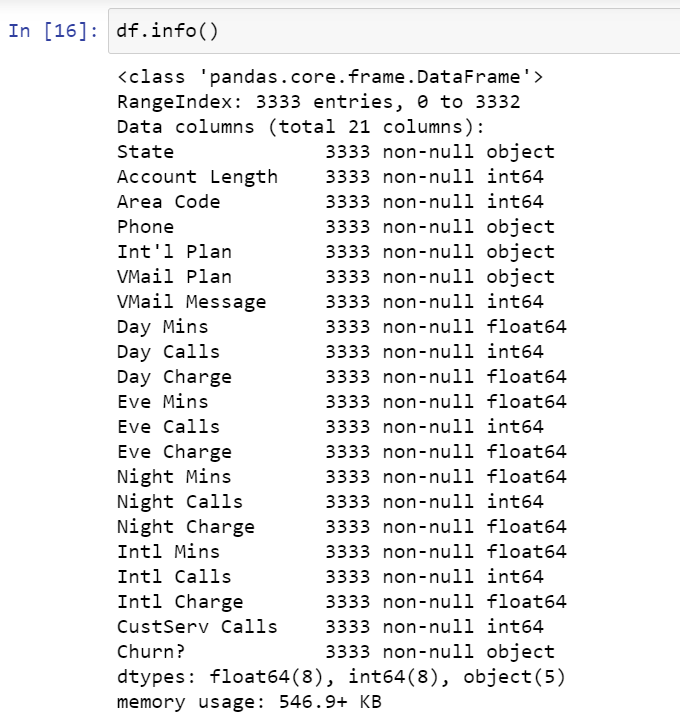




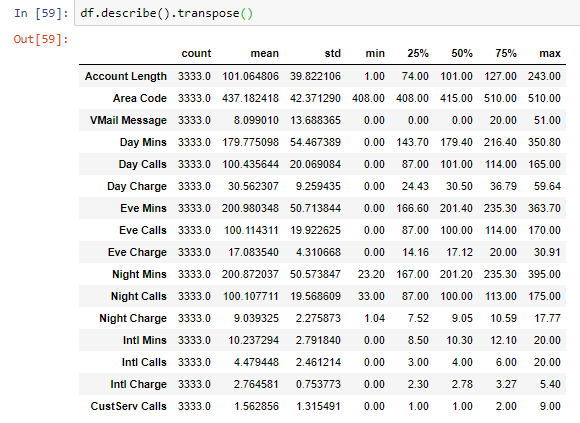
Thanks to pandas profiling we are able to identify the numerical statistics. Even though pandas profiling checks for missing values per attribute in the dataset we can check for missing values using the below command. As you can see this dataset provided seems pretty clean so far with no missing values in any of the attributes.



Now let’s look for datatypes within the dataset. With df.info() we can succinctly analyze the datatype in the dataset at the moment. We have 5 non-numeric attributes. One of them is the churn attribute, which is going to be our independent variable later.

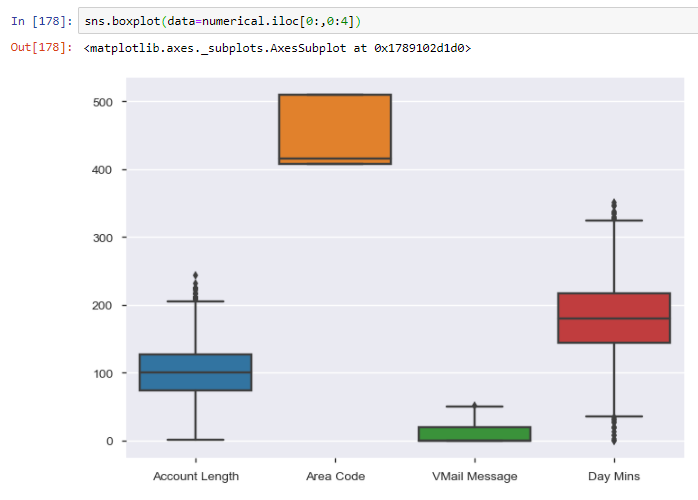


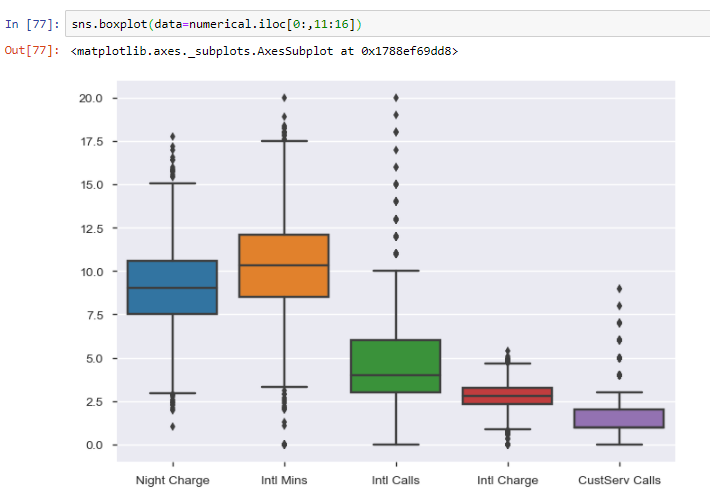
An alternative to the pandas profiling method is to us df.describe() function in pandas. This way we can quickly go through the attributes and find our max, min, mean and standard deviation of attributes.

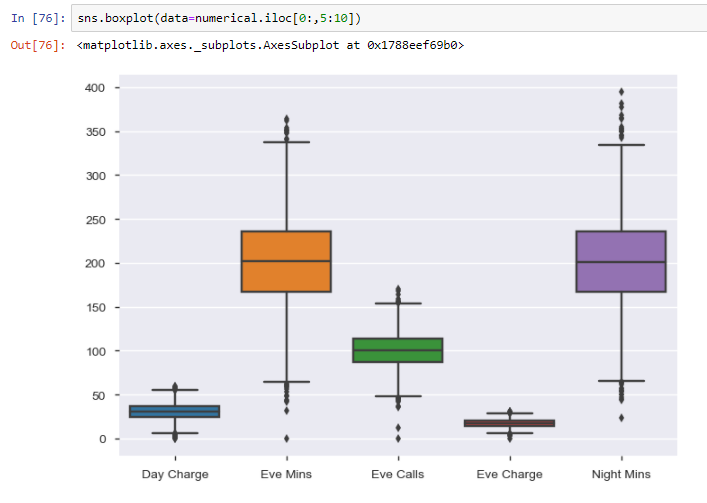


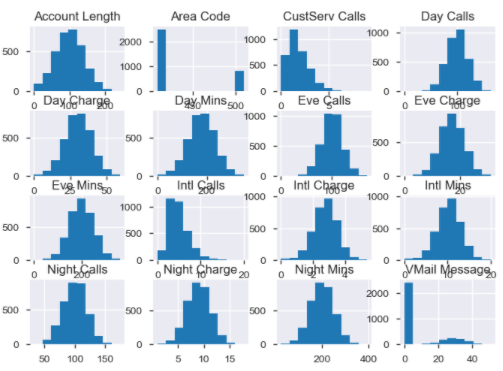


**3.3 Boxplots and Histograms**



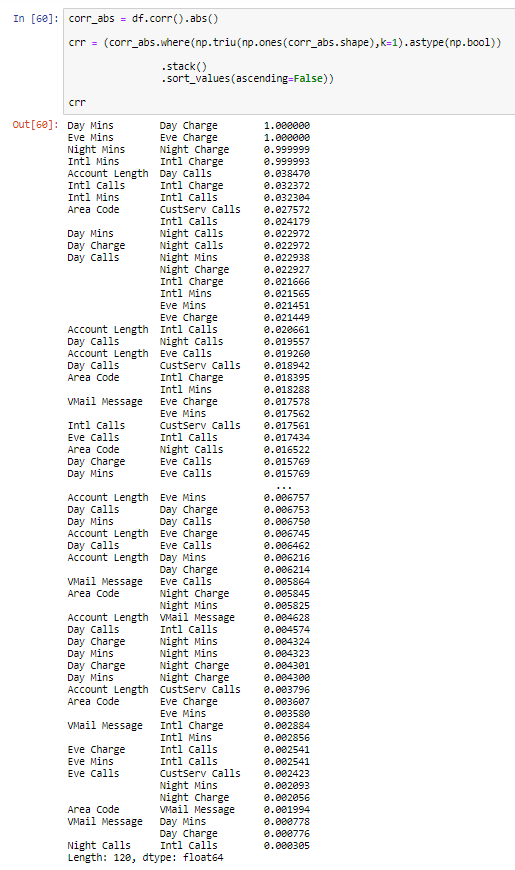






The dots within the box and whisker plots represents the outliers for that particular attribute. The program is calculating them by using this equation -> 1.5 +/- IQR[Q3-Q1]. If a dot is outside of Q1-1.5\*IQR or Q3+1.5\*IQR then it can be considered an outlier. We decided to keep the outliers for our study. As for the distributions most of the diagrams display a normal distribution with the exception of a few. The ones that display a skewness would be CustServ Calls and Intl Calls. Area code has an extremely high variance of area codes which is shy the graph looks very odd. Vmail message has a high number of clients not leaving a message which is why the graph looks so disproportionate.

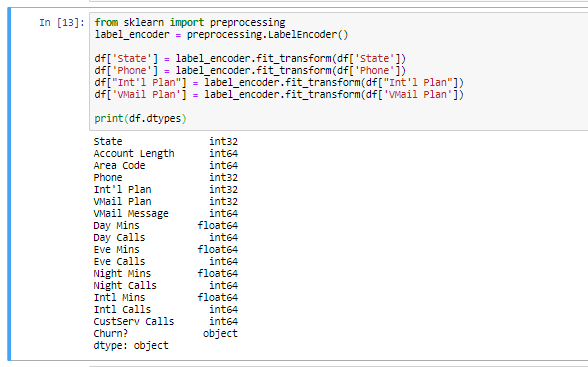
**3.4 Feature Selection**



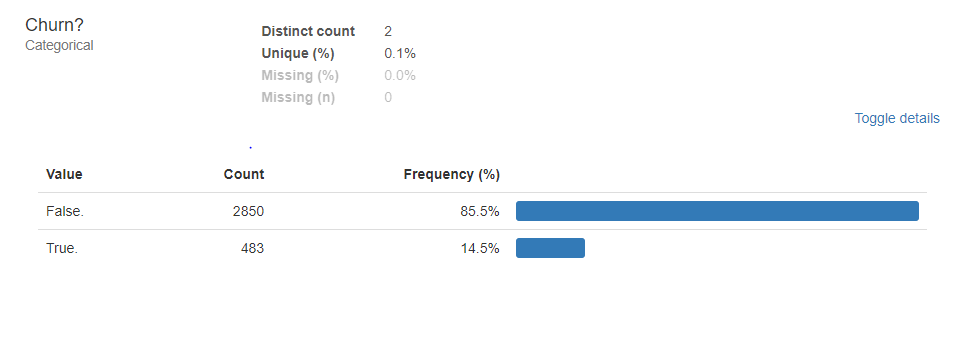
With Pandas profiling we can see that it rejects 4 attributes. These attributes are the charges which include day charge, night charge, international charge, and evening charge. The reason why we reject these attributes is because the correlation value is practically perfect or being as close to 1 as you can get. This means that really isn’t a reason to use because we already understand the relationship the charges represent; as time increased the charge increases. Therefore, there is no new information to gain from these attributes, which is why we reject them.



We later decided to convert the categorical variables into numerical with the exception of Churn. This is required when we run our classifiers because they won’t work with numerical attributes. Instead of using dummy variables we decided to use label encoder.

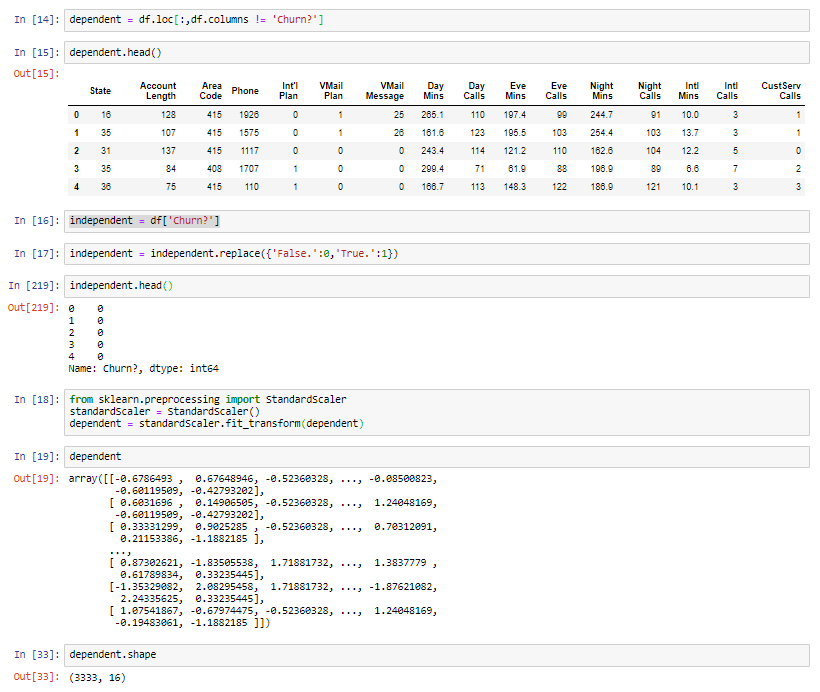


**3.5 Balancing the dataset**

In order for us to get a higher accuracy it’s important for us to balance the dataset. With an imbalanced dataset we may see some biases with different classifiers. The method we used to balance the data set was oversampling from smote. With oversampling we are essentially adding data to better represent our independent variable. This will be shown in the predictive modeling section.

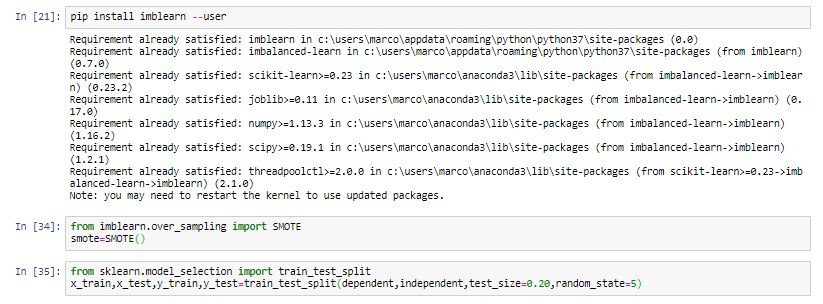
**3.6 Scaling the data**

Since we are still in the data preparation stage we are just going to prepare the dataset for balancing. First, we wanted to separate our independent and dependent variables. This is important since we will later use them for creating the split as well as balancing the dataset. We also converted the independent variable ‘Churn?’ into a numerical variable. We did this by using the replace function to assign False to 0 and True to 1. And then we scale the dependent variable so the data is all on the same scale. Unfortunately, we weren’t able to scale individual columns so we settled for making the entire dataset as an array.

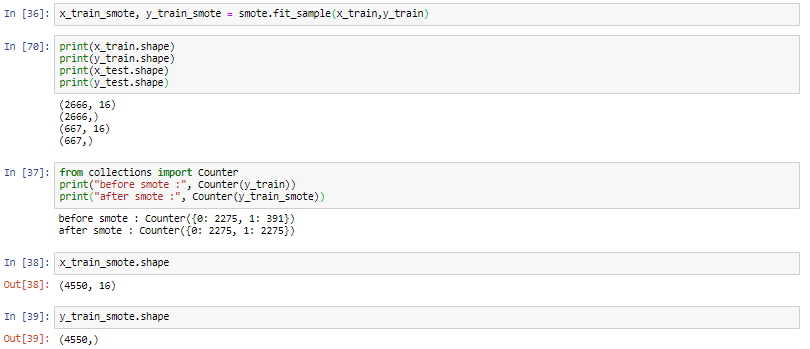


# 4. Predictive Modeling (Classification)

To split the data, we decided to use the train test split since it is a popular choice amongst most tutorials, articles and data scientists. We will be able to train and test our models this way to find the best classifier.



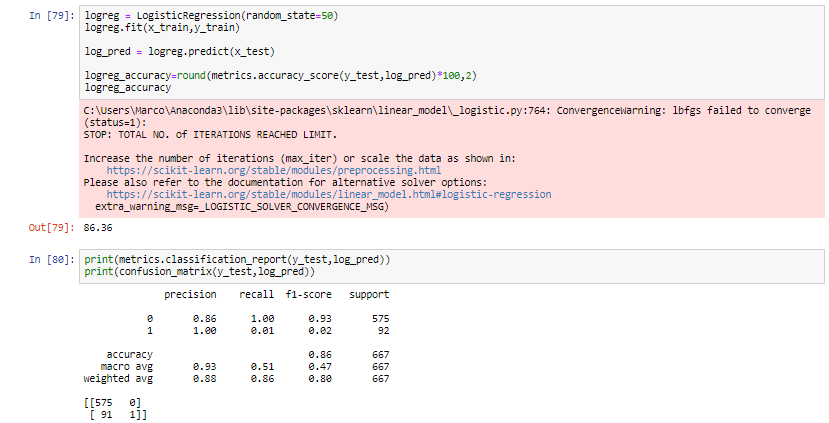
The type of split we use is the 80/20 split with a random state of five to select tuples at random. The testing data set will have 20 percent of the original dataset while the train dataset has the remainder. We will be addressing our imbalance prior to executing the classifiers with smote like we stated earlier.



After we apply oversampling, we notice that we have an increase in the number of rows of churned clients. It now matches the amount of people who did not churn in the original dataset.

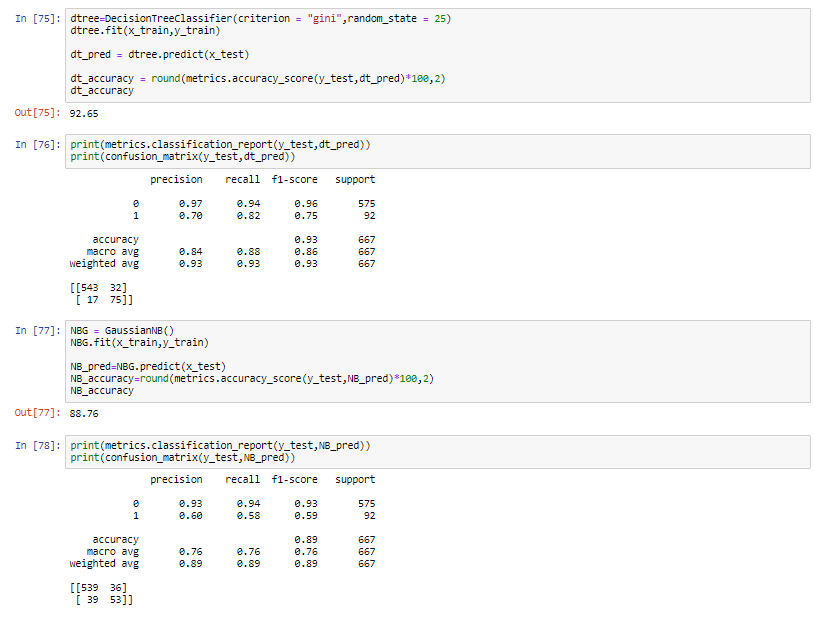
**4.1 Baseline**

The way we designed the baseline was relatively simple. We imported the data set, made the dataset readable, split the dataset and then we tested the data with the classifiers. Logistic Regression is used as the baseline model to compare Decision Tree and Naïve Bayes with to better understand the performance of the classification models. Logistic Regression has an accuracy of 86.36%. **Accuracy** is the ratio of correct predictions to total predictions made.



The confusion matrix shows that data was correctly predicted 576 times, while incorrectly predicted 91 times. We should also notice that the algorithm is unhappy that the data is not scaled and yielding very biased results. This is why the logistic regression model stopped its iterations in the above warning message.

We also ran the baseline model with Decision Tree and Naïve Bayes to further see the bias within the dataset. The models are overfitting and favouring with the event of not churning. The goal of our other models is to achieve the highes accuracy with our selected features and balanced dataset.



The classification report for each further provides the following:

1. **Precision**: The ability of a model to correctly predict positive cases

(Number of true positive cases) / (Number of *all the positive* cases)  
\*all the positive classes = true positive + false positive

The baseline model is accurately able to predict 86% of the first label and 100% for the second label.

The report shows that for accuracy the DT classifier is correctly able to predict 97% for the first label and 70% for the second label.

On the other hand, the NB classifier is correctly able to predict 93% and 60% for the second label.

1. **Recall:**

**Recall** quantifies the number of positive class predictions made out of all positive examples in the dataset. So, only corrected measured instances, which are true-positive and false-negatives, are concerned.  
(Number of true positives) / (# of true positives + # of false negatives)  
The baseline model’s recall rate is 100% for the first label and 1% for the second label.

The report shows that the Recall rate for the decision tree is 94% for the first label and 82% for the second.

The recall rate for NB is 94% for the first label and 58% for the second.

1. **F1-score**  
   This is a weighted harmonic mean value using both Precision and Recall. This measure is pretty useful when the dataset has an imbalanced distribution of different labels. It varies between 0-1 from minimum to maximum.  
   {(Precision \* Recall) \* 2} / (Precision + Recall)

The F1 score for the baseline model is 0.93 for the first label and 0.02 for the second label.

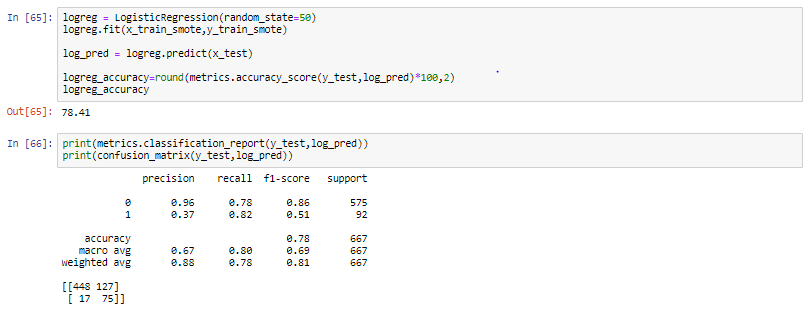
The F1 score for DT classifier is 0.96 for the first label and 0.75 for the second label. While in the case of NB it is 0.93 for the first label and 0.59 for the second.

1. **Support**  
   Support is the number of occurrences of each class label in the \*y\_test\* dataset. In the Churn dataset the first class label had 575 cases and the second had 92.

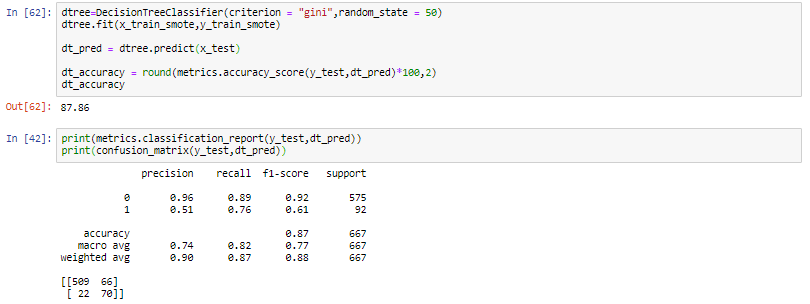
Both Naïve Bayes and Decision Tree are good performing classification algorithms for the Churn dataset in comparison to the selected baseline model of logistic regression. However, the decision tree has greater accuracy and is able to more correctly predict true positives and true negatives in comparison to Naïve Bayes.

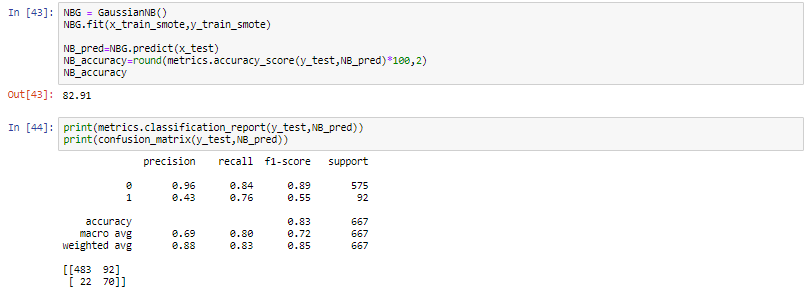
**4.2 Classifier Analysis.**

The logistic regression model is the weakest in terms of accuracy and metrics. The precision, recall and f1-score rank amongst the other classifiers.



The decision tree model is definitely the most accurate when the train model is applied to y\_test.





|  |  |
| --- | --- |
| **Classifier Model** | **Accuracy** |
| Decision Tree | 87.86 |
| Naïve Bayes | 82.91 |
| Logistic Regression | 78.41 |

# 5. Conclusion

We first started to clean the data and concluded that 16 variables where the best when it came to creating our model’s dependent variables and the churn column remained our independent variable.

We wanted to select 3 classifiers for comparing our models instead of just the standard 2. We selected Decision Tree, Naïve Bayes and Logistic Regression as our classifiers. In the end we decided that decision tree was the most accurate due to it having the highest values when it came to all metrics and accuracy of the model. In comparison to the baseline, the accuracy is lower due to the lack of scaling in the dataset. This is because the baseline model has more noise within the model since it has the attributes we decided to remove in our more complex model.

We would advise the company to use the decision tree model to predict whether a client would consider churning or not. It also important for the