**REPORT FROM FINAL PROJECT – INTRODUCTION TO MACHINE LEARNING**

**COURSE: DATA SCIENCE FOUNDATIONS II**

**STUDENT: LEANDRO BARUCH**

**Identify Fraud from Enron Dataset**

***“Enron Corporation****was an*[*American*](https://en.wikipedia.org/wiki/United_States)[*energy*](https://en.wikipedia.org/wiki/Energy_development)*,*[*commodities*](https://en.wikipedia.org/wiki/Commodities)*, and services company based in*[*Houston*](https://en.wikipedia.org/wiki/Houston)*,*[*Texas*](https://en.wikipedia.org/wiki/Texas)*. It was founded in 1985 as a merger between*[*Houston Natural Gas*](https://en.wikipedia.org/wiki/Houston_Natural_Gas)*and [InterNorth](https://en.wikipedia.org/wiki/InterNorth" \o "InterNorth), both relatively small regional companies. Before its bankruptcy on December 3, 2001, Enron employed approximately 29,000 staff and was a major*[*electricity*](https://en.wikipedia.org/wiki/Electricity)*,*[*natural gas*](https://en.wikipedia.org/wiki/Natural_gas)*, communications and*[*pulp and paper*](https://en.wikipedia.org/wiki/Pulp_and_paper_industry)*company, with claimed revenues of nearly $101 billion during 2000.*[*Fortune*](https://en.wikipedia.org/wiki/Fortune_(magazine))*named Enron "America's Most Innovative Company" for six consecutive years.*

*At the end of 2001, it was revealed that Enron's reported financial condition was sustained by institutionalized, systematic, and creatively planned*[*accounting fraud*](https://en.wikipedia.org/wiki/Accounting_scandals)*, known since as the*[*Enron scandal*](https://en.wikipedia.org/wiki/Enron_scandal)*. Enron has since become a well-known example of willful corporate*[*fraud*](https://en.wikipedia.org/wiki/Fraud)*and*[*corruption*](https://en.wikipedia.org/wiki/Corporate_crime)*. The scandal also brought into question the accounting practices and activities of many corporations in the*[*United States*](https://en.wikipedia.org/wiki/United_States)*and was a factor in the enactment of the*[*Sarbanes–Oxley Act*](https://en.wikipedia.org/wiki/Sarbanes%E2%80%93Oxley_Act)*of 2002. The scandal also affected the greater business world by causing the dissolution of the*[*Arthur Andersen*](https://en.wikipedia.org/wiki/Arthur_Andersen)*accounting firm, which had been Enron's main auditor for years.*

*Enron filed for bankruptcy in the*[*Southern District of New York*](https://en.wikipedia.org/wiki/Southern_District_of_New_York)*in late 2001 and selected*[*Weil, Gotshal & Manges*](https://en.wikipedia.org/wiki/Weil,_Gotshal_%26_Manges)*as its bankruptcy counsel. It ended its bankruptcy during November 2004, pursuant to a court-approved plan of reorganization. A new board of directors changed the name of Enron to****Enron Creditors Recovery Corp****., and emphasized reorganizing and liquidating certain operations and assets of the pre-bankruptcy Enron. On September 7, 2006, Enron sold*[*Prisma Energy International Inc.*](https://en.wikipedia.org/wiki/Prisma_Energy_International)*, its last remaining business, to Ashmore Energy International Ltd. (now AEI).” – From Wikipedia, the free encyclopedia*

The objective of this project, is use both financial and email data from 146 executives, and use those data to identify Persons of Interest (POIs). For that, I will analyse the dataset, all the data features, understand their correlations and see which are the best for this purpose. After that I will create a Machine Learning Model to identify the POIs based on those features and check their efficiency.

Summary

[**Exploring the Dataset** 2](#_Toc533411581)

[**Features with missing data** 4](#_Toc533411582)

[**Checking the Features** 5](#_Toc533411583)

[**POIs and Not POIs comparison** 6](#_Toc533411584)

[**Metrics** 7](#_Toc533411585)

[**Checking Classifiers with all features** 8](#_Toc533411586)

[**Checking Classifiers with 5 top features** 9](#_Toc533411587)

[**Checking Classifiers with 3 top features** 9](#_Toc533411588)

[**Checking Classifiers with the top features** 10](#_Toc533411589)

[**Checking Classifiers with 7 top features** 11](#_Toc533411590)

[**Checking Classifiers with 10 top features** 12](#_Toc533411591)

[**Checking Classifiers with new feature** 12](#_Toc533411592)

[**Checking Classifiers with top 7 features from SelectKBest** 13](#_Toc533411593)

[**Checking DecisionTree and RandomForest with adjusted parameters using GridSearchCV** 14](#_Toc533411594)

[**Why all the tests?** 17](#_Toc533411595)

[**Validation** 17](#_Toc533411596)

[**Conclusion** 17](#_Toc533411597)

[**References** 17](#_Toc533411598)

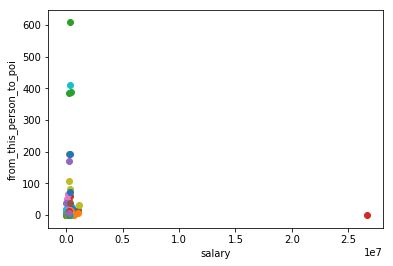
# **Exploring the Dataset**

During dataset exploration, I have found some interesting information:

* We have 146 registers in our dataset, one of them was the ‘TOTAL’, so it was removed and now we have 145 registers;
* From 145 people, 18 of them are flagged as POI, they are:
  + Hannon Kevin P, Colwell Wesley, Rieker Paula H, Kopper Michael J, Shelby Rex, Delainey David W, Lay Kenneth L, Bowen Jr Raymond, Belden Timothy N, Fastow Andrew S, Calger Christopher F, Rice Kenneth D, Skilling Jeffrey K, Yeager F Scott, Hirko Joseph, Koenig Mark E, Causey Richard A and Glisan JR Ben F;
* The features we have at our disposal are:
  + salary, to\_messages, deferral\_payments, total\_payments, exercised\_stock\_options, bonus, restricted\_stock, shared\_receipt\_with\_poi, restricted\_stock\_deferred, total\_stock\_value, expenses, loan\_advances, from\_messages, other, from\_this\_person\_to\_poi, poi, director\_fees, deferred\_income, long\_term\_incentive, email\_address, from\_poi\_to\_this\_person

Checking those features, of course those which more call attention are Salary, Bonus, Total Stock Value. These features can tell us which person are receiving more money from Enron, and outliers may be guilty in this case.

**Check #1**

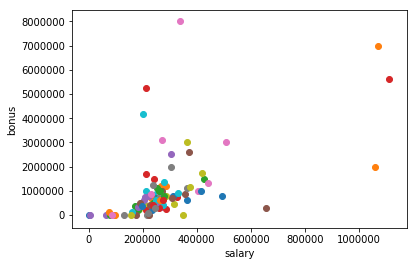


*Salary x Messages to POI*

We can see a big outlier.

After checking it, its name is ‘TOTAL’. Probably when the dataset was created someone forgot to delete the Total Row. I just deleted it.

**Check #2**

****

*Salary x Bonus*

Now we can see 4 more outliers. Three of them with the higher salaries in company, and one with the highest bonus.

The person with the highest bonus, is Lavorato John J, head of operations with a bonus of $8.000.000. And he is not flagged as POI. Suspicious…

The three top salaries of Enron are:

* Skilling Jeffrey K, CEO
  + Salary: $1.111.258
* Lay Kenneth L, founder and CEO
  + Salary: $1.072.321
* Frevert Mark A, CEO
  + Salary: $1.060.932

Those three occupied the highest role in the company, but it means that big difference in salary, since we are talking only about the high level of executives of Enron? Also, Frevert Mark A was not flagged as POI.

Seeing those differences, make me wonder how much is the average salary and bonus for high executive level in Enron. And the result is:

* Average salary: $ 284.088
* Average bonus: $ 1.201.773

It means the examples above are way higher the average.

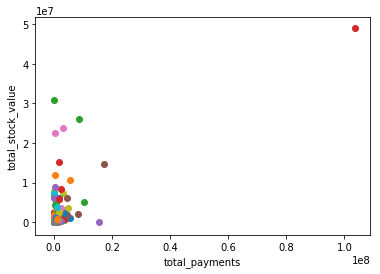
Also, to be honest, this average salary looks weird. The Total was around $26kk and dividing it per number of people, I expected another value.

So, I have checked for ‘NaN’ values and I have found this:

* Amount of NaN salary: 51
* Amount of NaN bonus: 64

Looks like the data is far from complete and it may interfere the capability of our model detect new POIs. We will have to use the data from e-mails to be more precise.

**Check #3**

****

*Total Payments x Total Stock Value*

Now, considering all payments and stock value, a big outlier appears. And it is his second time, Lay Kenneth L with the big value of $103.559.793 in total payments and $49.110.078 of total stock value.

# **Features with missing data**

As we see in Check#2, salary and bonus have a lot of missing values. Let’s check if we have other cases with other features.

After a check, there is a lot of missing data in our dataset, below the results:

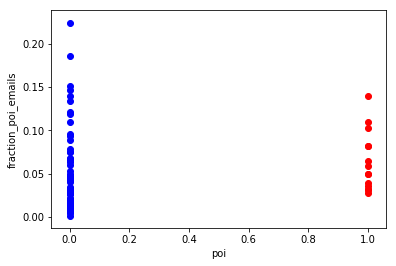
* Amount of NaN to\_messages: 59
* Amount of NaN deferral\_payments: 107
* Amount of NaN total\_payments: 21
* Amount of NaN exercised\_stock\_options: 44
* Amount of NaN restricted\_stock: 36
* Amount of NaN shared\_receipt\_with\_poi: 59
* Amount of NaN restricted\_stock\_deferred: 128
* Amount of NaN total\_stock\_value: 20
* Amount of NaN expenses: 51
* Amount of NaN loan\_advances: 142
* Amount of NaN from\_messages: 59
* Amount of NaN other: 53
* Amount of NaN from\_this\_person\_to\_poi: 59
* Amount of NaN director\_fees: 129
* Amount of NaN deferred\_income: 97
* Amount of NaN long\_term\_incentive: 80
* Amount of NaN from\_poi\_to\_this\_person: 59

Some features are expected to have missing values like ‘loan\_advances’, ‘long\_term\_incentives’, ‘from\_this\_person\_to\_poi’ and ‘from\_poi\_to\_this\_person’, but I see it’s a big amount event to other features. It can impact in the performance of our models, that’s why we will need to do various tests to find the best scenario.

# **Checking the Features**

Now we have removed an outlier and detected some behaviors from dataset, we need to check the features, and see what feature is more important in detecting a POI.

As I said before, use the amount of e-mails sent and received from POIs may be a good indicative to find a POI. Also, we have a few examples that received a big amount of cash but it’s not a POI.



As we can see with the green line, POIs tend to exchange e-mail among them at least 3% of their total e-mails while not POIs sometimes have 0%.

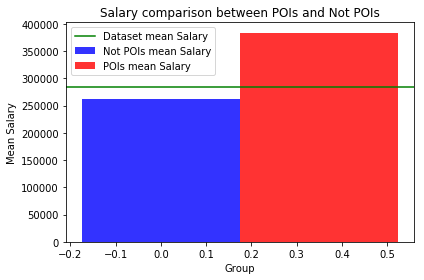
Also, we have a considerable difference between their averages:

POIs use to have an average of 5% of e-mails exchanged among them, while Not POIs uses to have an average of 2,5% of e-mails exchanged with POIs.

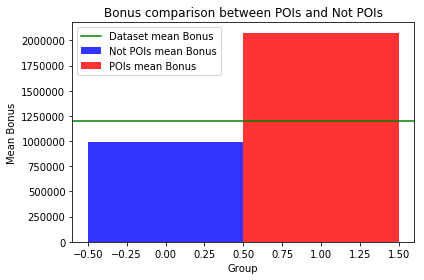
It’s worth to use this feature in our classifier to see if improves it.

# **POIs and Not POIs comparison**

According the information we got earlier regards the average salary and bonus, lets check in a plot, how is the difference of average salary and bonus considering the two groups.



As we can see in the image above, the mean salary from people flagged as POIs is way higher than Not POIs and the dataset mean salary.



Now using bonus information. The difference is even higher in favour of POIs. Looks like fat bonus is a good way to convince people to change side.

The conclusion after seeing those informations is that the people flagged as POI was receiving much more money than the not POIs.

In numbers, we have:

***Salary difference from POI and Not POI: 1.46***

***Bonus difference from POI and Not POI: 2.10***

# **Metrics**

First, I believe we have too much features for the training points we have.

Let’s check:





We will be looking for the training and prediction time that takes to created the model and some metrics;

**Accuracy**: is the fraction of predictions our model got right. Accuracy has the following formula:

**Precision:** proportion of positive identifications was actually correct. Precision has the following formula:

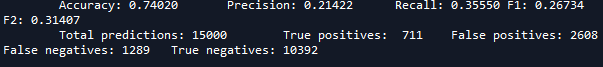
**Recall:** proportion of actual positives was identified correctly. It’s formula is:

**F1 Score:** it is the balance between Precision and Recall. The F1 Score formula is:

**The performance requested for this project, is Precision and Recall above 0.30**

# **Checking Classifiers with all features**

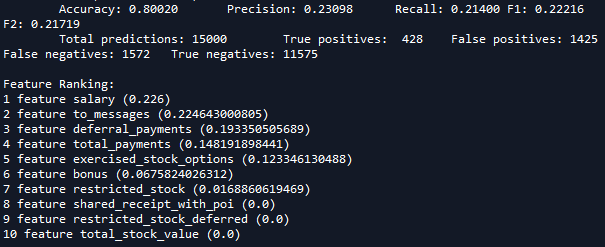
**SVM LinearSVC** (failed)



**Naive Bayes** (failed)



**Decision Tree** (failed)



**Random Forest** (failed)

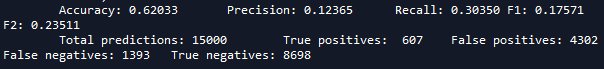


So, our F1 Score is too low as expected. Let’s use the more important features from Decision Tree and check again.

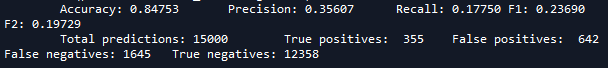
# **Checking Classifiers with 5 top features**

The five features that we will use in this test are: salary, to messages, deferral payments, total payments and exercised stock options.

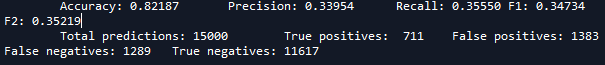
**SVM LinearSVC** (failed)



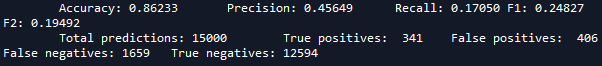
**Naive Bayes** (failed)



**Decision Tree –** (approved)



**Random Forest** (failed)



# **Checking Classifiers with 3 top features**

The three features that we will use in this test are: salary, to messages and deferral payments.

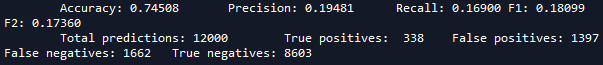
**SVM LinearSVC** (failed)



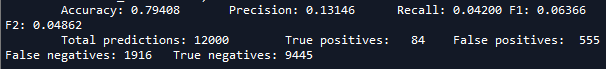
**Naive Bayes** (failed)



**Decision Tree** (failed)



**Random Forest** (failed)

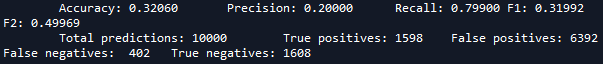


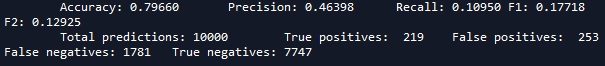
No good results came from 3 features. Next scenario will be using the top feature, salary.

# **Checking Classifiers with the top features**

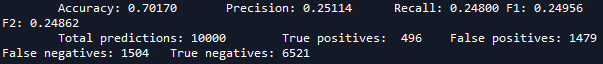
The top feature that we will use in this test is salary.

**SVM LinearSVC** (failed)

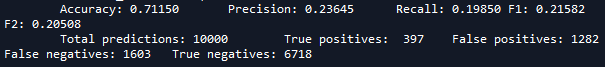


**Naive Bayes** (failed)

**Decision Tree** (failed)



**\Random Forest** (failed)

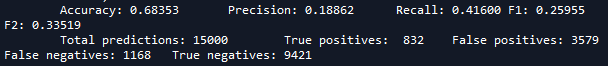


We could see in our last two scenarios, that decreasing the number of features didn’t work well. Let’s try using the 7 best features from ‘DecisionTree feature\_importances\_’

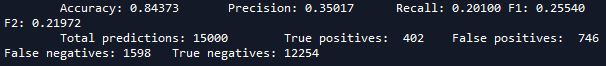
# **Checking Classifiers with 7 top features**

The seven features that we will use in this test are: salary, to messages, deferral payments, total payments, exercised stock options, bonus and restricted stock.

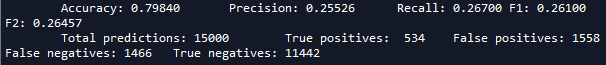
**SVM LinearSVC** (failed)



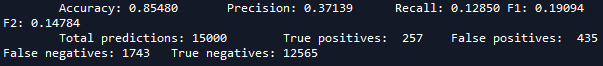
**Naive Bayes** (failed)



**Decision Tree** (failed)



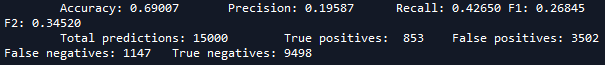
**Random Forest** (failed)



# **Checking Classifiers with 10 top features**

The ten features that we will use in this test are: salary, to messages, deferral payments, total payments, exercised stock options, bonus and restricted stock.

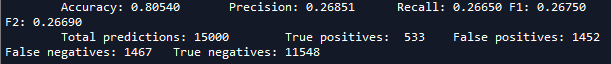
**SVM LinearSVC** (failed)



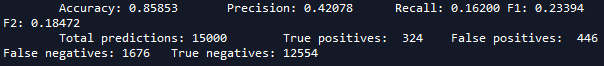
**Naive Bayes** (failed)



**Decision Tree** (failed)



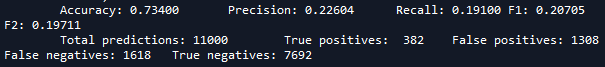
**Random Forest** (failed)



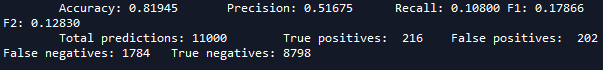
# **Checking Classifiers with new feature**

In this scenario, we will use poi plus our top 1 feature and our new feature. So, we will have: fraction poi emails and salary.

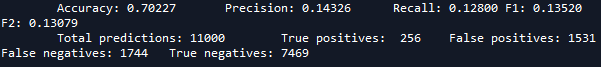
**SVM LinearSVC** (failed)



**Naive Bayes** (failed)



**Decision Tree** (failed)



**Random Forest** (failed)



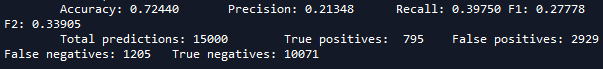
# **Checking Classifiers with top 7 features from SelectKBest**

As we saw, the new feature didn’t improve our results, what did was using the top 7 features. We had a good result with SVM and Naive Bayes, but I believe we can get even better, maybe using SelectKBest.

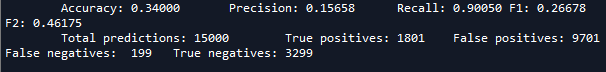
After user SelectKBest, I get to another top 7 of features: deferra\_payments, total\_payments, exercised\_stock\_options, bonus, restricted\_stock\_deferred, director\_fees, deferred\_income

Lets try again and see if we get some improvement.

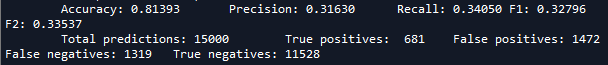
**SVM LinearSVC** (fail)



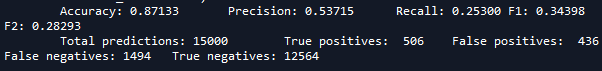
**Naive Bayes** (fail)



**Decision Tree** (approved)



**Random Forest** (fail)



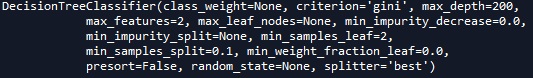
Considering all tests made, Decision Tree was approved in 2 scenarios, using top 5 features and the top 7 features selected by SelectKBest.

Also, RandomForest had promising results, and maybe tuning it with GridSearchCV we can reach our objective

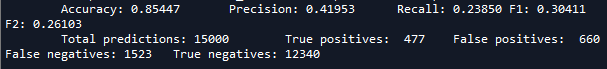
# **Checking DecisionTree and RandomForest with adjusted parameters using GridSearchCV**

**DecisionTree final test**

The parameters selected are:



The result was next to using default parameters but worse Recall, as you can see below:

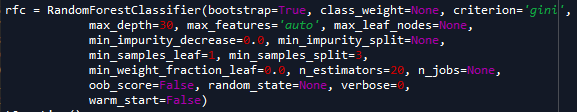


**DecisionTree Conclusion**

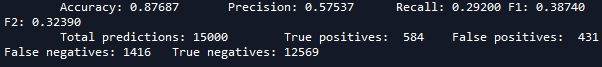
Our best result was using Top 5 features from DecisionTree, and default parameters as you can see in the Overall Table

**RandomForest final test**

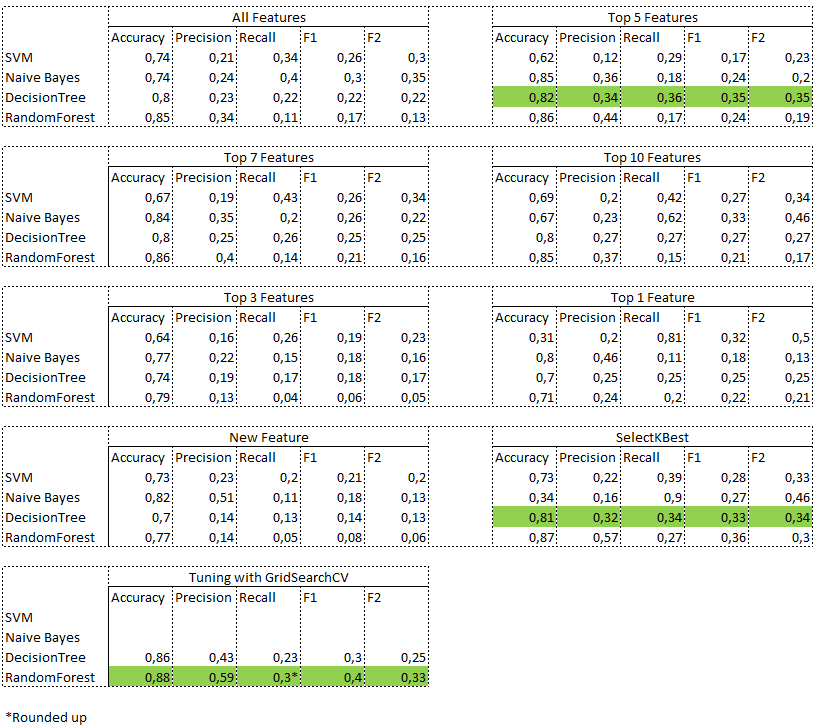
Parameters suggested by GridSearchCV:



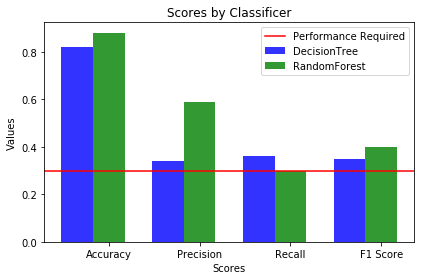
Our result was improved and approved by the rounded Recall value. The other results are better than DecisionTree, but since RandomForest was approved because the rounded value, I will choose DecisionTree as our best classifier.



**Check the table of Overall Results**



Below a plot comparing best result from DecisionTree and RandomForest



# **Why all the tests?**

We did a lot of tests, considering all features, features selected by DecisionTree.features\_importances\_, and SelectKBest. The reason to that, is because we need consider many variables that can occur with our dataset. For example, the number of features can be overfitting, which means we have more features than necessary, some of them may decrease of result because of missing data or even conflict with another feature. Also, we have the classifier parameters, which GridSearchCV help us to find the best variation in an automatic manner.

All of that steps are needed to insure the result with more proximity from reality with the given data.

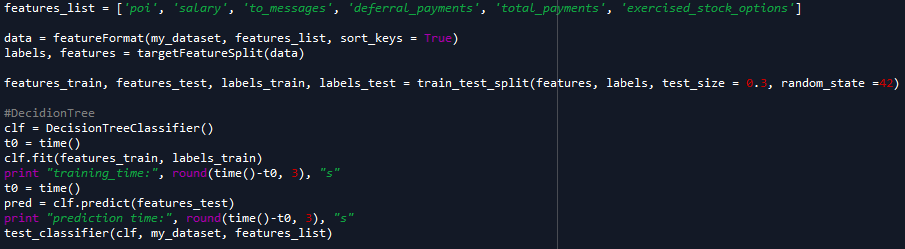
# **Validation**

Our validation model, creates 1000 combinations of train and test values.

It’ is very important, because when we split our train and test data, we can have many different variations. Creating a scenario that is possible to test it with 1000 different combinations and use the average results can guarantee a fine result of our model considering the most of situations possible.

# **Conclusion**

Best result was using DecisionTree with those data below:



# **References**

I used the site <https://scikit-learn.org> to check parameters from classifiers and my own mini projects from this course.

I used the Git Hub from user zelite to find a function to calculate my new feature: <https://github.com/zelite>

I used ProgramCreek website to get examples of how to use SelectKBest: <https://www.programcreek.com/python/example/93974/sklearn.feature_selection.SelectKBest>