Utilizing Machine Learning to Identify Fraudulent Activities through Analysis of Financial and Communication Data

Group 12
Presenter: Lin Xiaojian

Group Member
Lin Xiaojian (Leader)
Li Zhaochen
Liang Xinwen
Liu Jialin
Yan Shaochong

Table of Content

01 Project Goal

02 Exploring the Dataset

03 Feature Selection

04 Determine Algorithm

05 Validation and Evaluation

06 Conclusion

01 Project Goal

Project Goal

Project Objective

Dataset Objective

Analysis Process

Model Evaluation and Optimization

Background

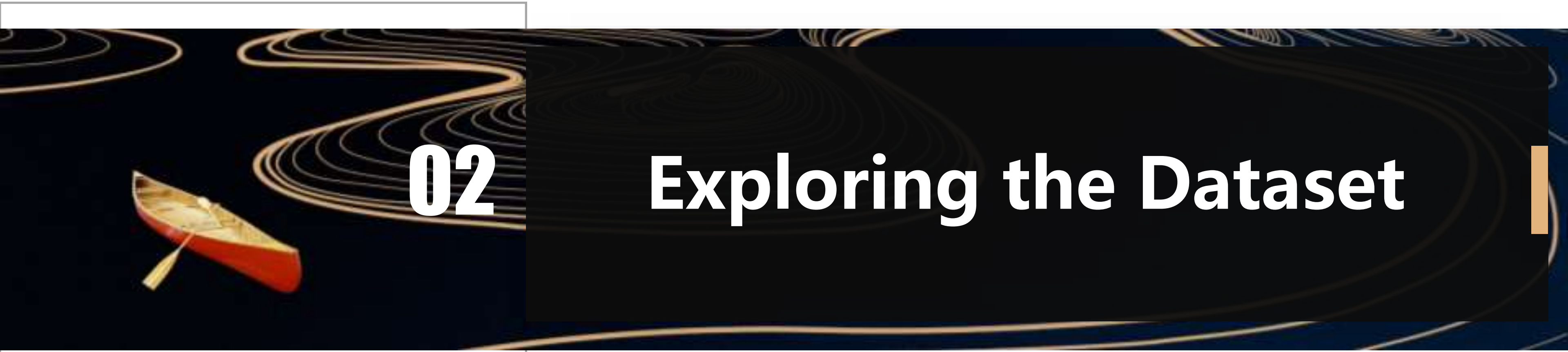
Why Enron?

In late 2001, Enron, an American energy company, filed for bankruptcy after one of the largest financial scandals in corporate history. After the company's collapse, over 600,000 emails generated by 158 Enron employees - now known as the Enron Corpus.

Today, the Enron Corpus is the largest and one of the only publicly available mass collections of data easily accessible for study.

Why Machine Learning?

> “ Machine Learning is an incredibly effective field when it comes to making predictions from data, especially large amounts of it. For example, the Enron Corpus, after cleaning, has around 500,000 emails, and attempting to identify fraudulent employees by manually foraging through half a million emails is a daunting task at best. ”



02 Exploring the Dataset

21 Features

- ['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']

- In the features, the boolean **poi** denotes whether the person is a person of interest or not. It is the most important variable. The Poi_count function shows that there are 18 such people in the dataset, and the aim of this project is to find distinguishing features that set these people apart from the others.

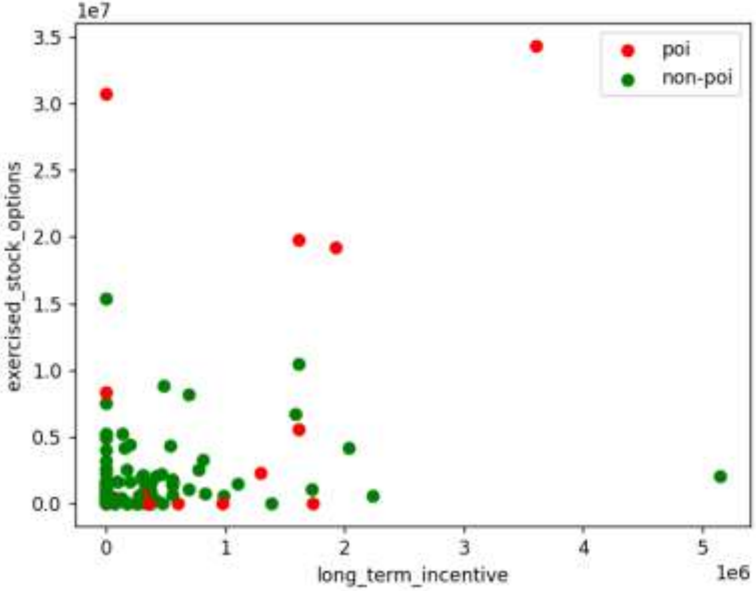
- Of course, not everyone has data for each feature, and missing data is denoted by 'NaN'. The NaN_count function prints a dictionary sorted in descending order.

03 Feature Selection

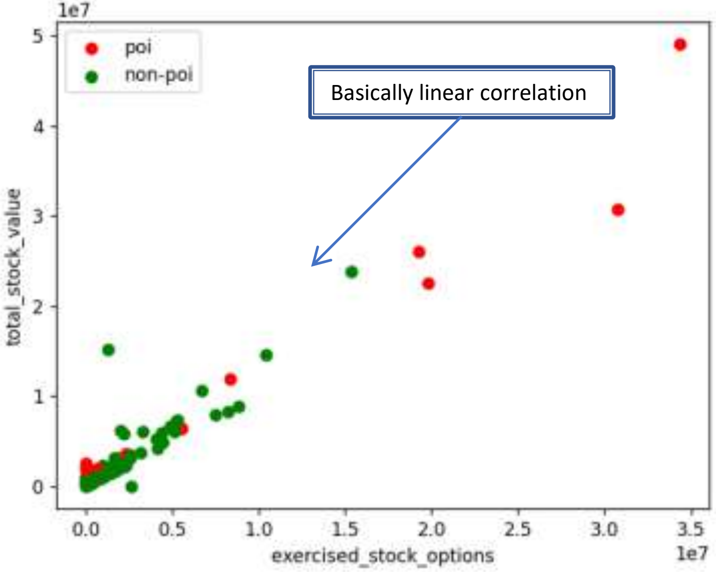
Visualizing Data

Simply create a graph of any two features and visualize them, looking for distribution and getting a general idea of how the data changes. This is a good way to show the relationship between variables.

exercised_stock_options vs long_term_Incentive

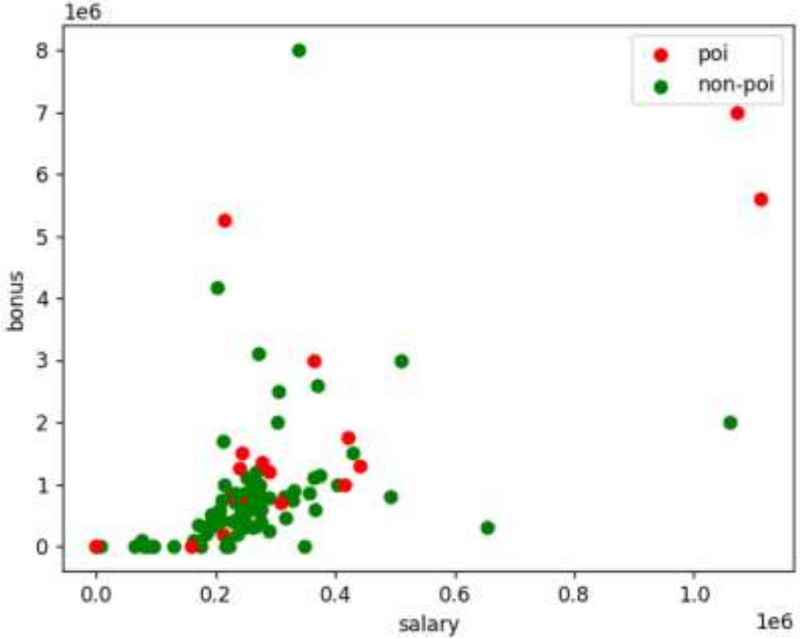


total_stock_value vs exercised_stock_options

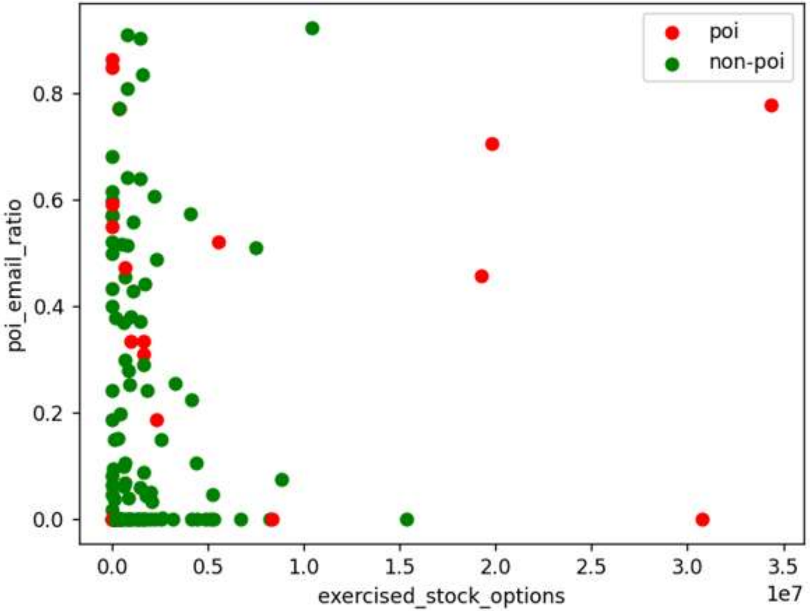

Basically linear correlation

Visualizing Data

salary vs bonus

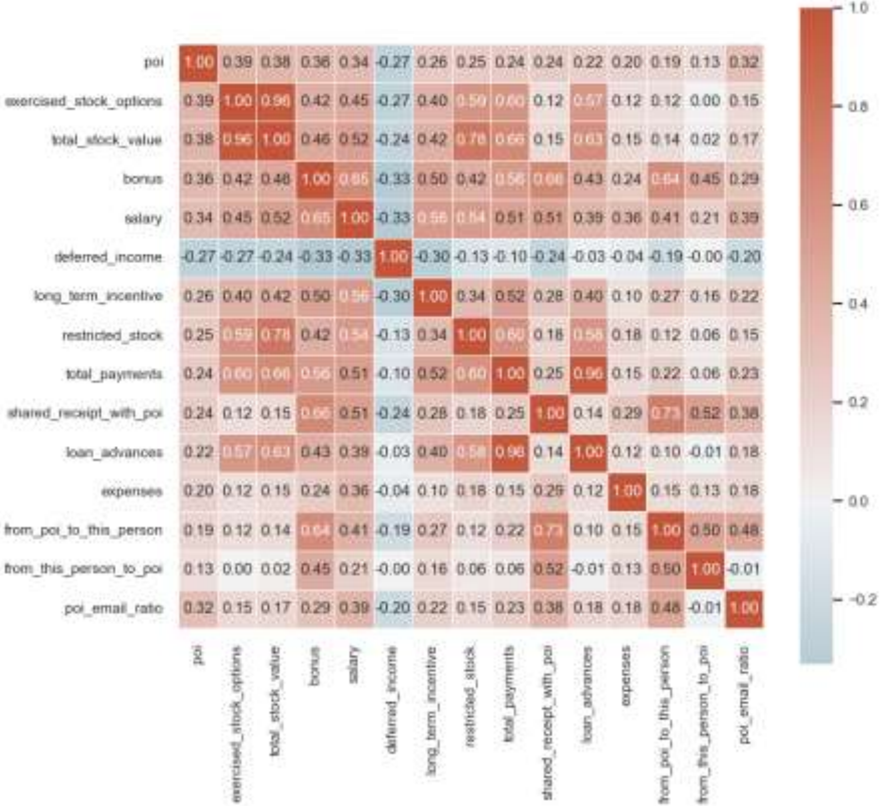


exercised_stock_options vs poi_email_ratio



Visualizing Data

Heatmap of Correlation Coefficients of all Feature Terms



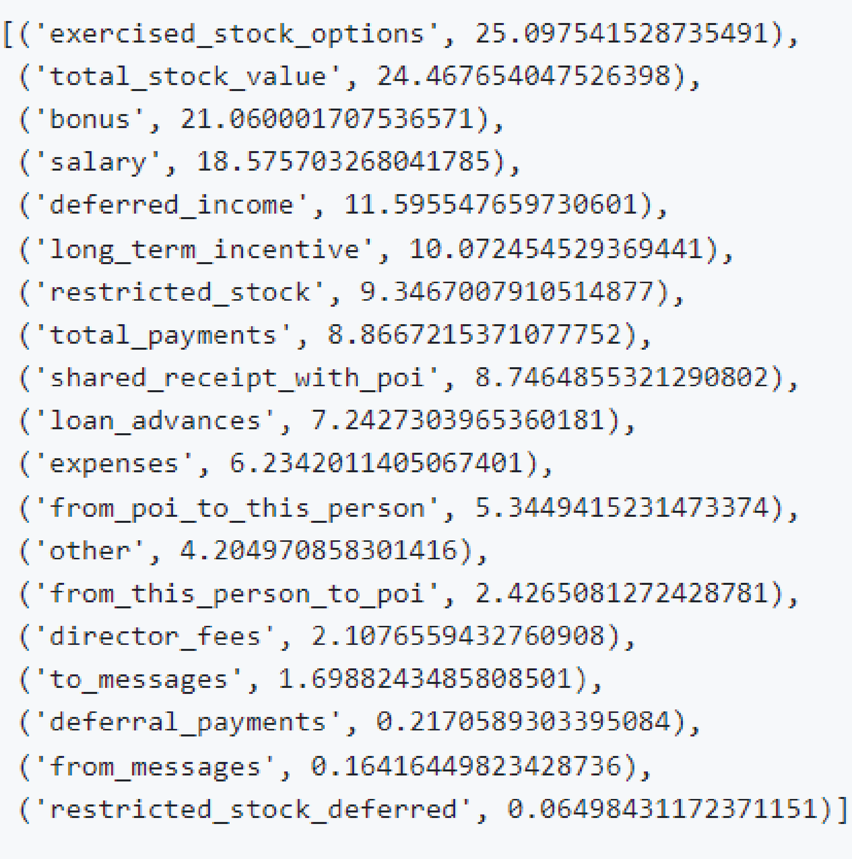
	poi	exercised_stock_options	total_stock_value	bonus	salary	deferred_income	long_term_incentive	restricted_stock	total_payments	shared_receipt_with_poi	loan_advances	expenses	from_poi_to_this_person	from_this_person_to_poi	poi_email_ratio
poi	1.00	0.39	0.38	0.36	0.34	-0.27	0.26	0.25	0.24	0.24	0.22	0.20	0.19	0.13	0.32
exercised_stock_options	0.39	1.00	0.96	0.42	0.45	-0.27	0.40	0.59	0.60	0.12	0.57	0.12	0.12	0.00	0.15
total_stock_value	0.38	0.96	1.00	0.46	0.52	-0.24	0.42	0.78	0.66	0.15	0.63	0.15	0.14	0.02	0.17
bonus	0.36	0.42	0.46	1.00	0.65	-0.33	0.50	0.42	0.56	0.68	0.43	0.24	0.64	0.45	0.29
salary	0.34	0.45	0.52	0.65	1.00	-0.33	0.56	0.54	0.51	0.51	0.39	0.36	0.41	0.21	0.39
deferred_income	-0.27	-0.27	-0.24	-0.33	-0.33	1.00	-0.30	-0.13	-0.10	-0.24	-0.03	-0.04	-0.19	-0.00	-0.20
long_term_incentive	0.26	0.40	0.42	0.50	0.56	-0.30	1.00	0.34	0.52	0.28	0.40	0.10	0.27	0.16	0.22
restricted_stock	0.25	0.59	0.78	0.42	0.54	-0.13	0.34	1.00	0.60	0.18	0.58	0.18	0.12	0.06	0.15
total_payments	0.24	0.60	0.66	0.56	0.51	-0.10	0.52	0.60	1.00	0.25	0.96	0.15	0.22	0.06	0.23
shared_receipt_with_poi	0.24	0.12	0.15	0.68	0.51	-0.24	0.28	0.18	0.25	1.00	0.14	0.29	0.73	0.52	0.38
loan_advances	0.22	0.57	0.63	0.43	0.39	-0.03	0.40	0.58	0.96	0.14	1.00	0.12	0.10	-0.01	0.18
expenses	0.20	0.12	0.15	0.24	0.36	-0.04	0.10	0.18	0.15	0.29	0.12	1.00	0.15	0.13	0.18
from_poi_to_this_person	0.19	0.12	0.14	0.64	0.41	-0.19	0.27	0.12	0.22	0.73	0.10	0.15	1.00	0.50	0.48
from_this_person_to_poi	0.13	0.00	0.02	0.45	0.21	-0.00	0.16	0.06	0.06	0.52	-0.01	0.13	0.50	1.00	-0.01
poi_email_ratio	0.32	0.15	0.17	0.29	0.39	-0.20	0.22	0.15	0.23	0.38	0.18	0.18	0.48	-0.01	1.00

Select K Best and Feature Creation

Our first step is to check the ability of each feature in clearly differentiating between POI and non-POI. To do this, we are going to use Scikit-Learn's SelectKBest algorithm, which will give me a score for each feature in it's ability to identify the target variable.

In select_k_best.py, the Select_K_Best function returns an array of k tuples in descending order of its score. Running this will show the most useful features and the not-so-useful ones. Running them over all features gives:

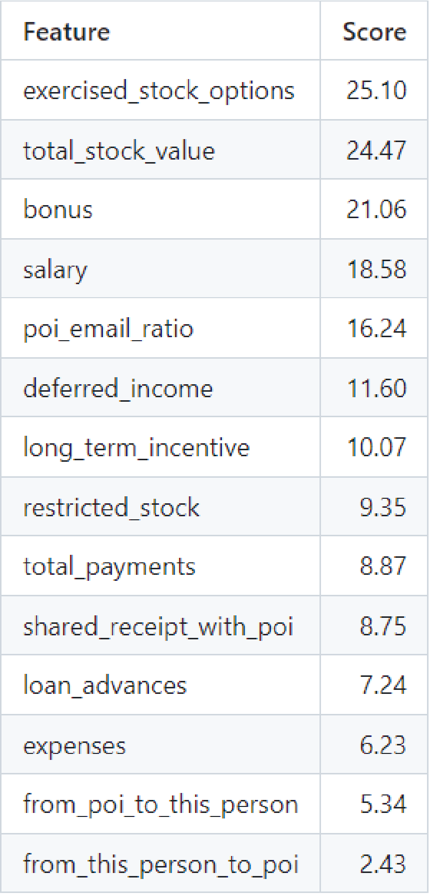
The first thing we notice is that ‘other’ is not very useful and also ambiguous, so it‘s not a feature we are going to add to the list.



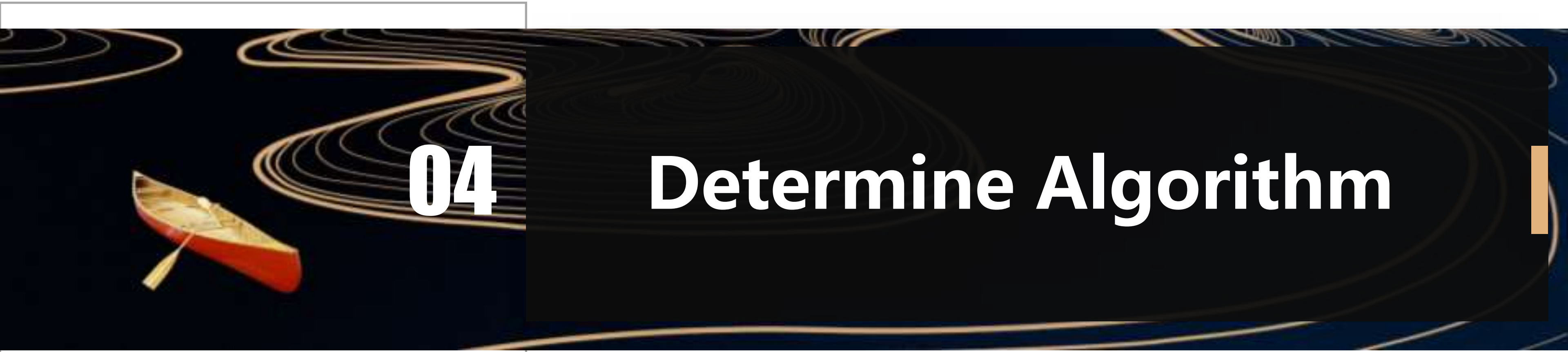
```
[('exercised_stock_options', 25.097541528735491),
 ('total_stock_value', 24.467654047526398),
 ('bonus', 21.060001707536571),
 ('salary', 18.575703268041785),
 ('deferred_income', 11.595547659730601),
 ('long_term_incentive', 10.072454529369441),
 ('restricted_stock', 9.3467007910514877),
 ('total_payments', 8.8667215371077752),
 ('shared_receipt_with_poi', 8.7464855321290802),
 ('loan_advances', 7.2427303965360181),
 ('expenses', 6.2342011405067401),
 ('from_poi_to_this_person', 5.3449415231473374),
 ('other', 4.204970858301416),
 ('from_this_person_to_poi', 2.4265081272428781),
 ('director_fees', 2.1076559432760908),
 ('to_messages', 1.6988243485808501),
 ('deferral_payments', 0.2170589303395084),
 ('from_messages', 0.16416449823428736),
 ('restricted_stock_deferred', 0.06498431172371151)]
```

Select K Best and Feature Creation

The final 14 features to be used for the machine learning process are:



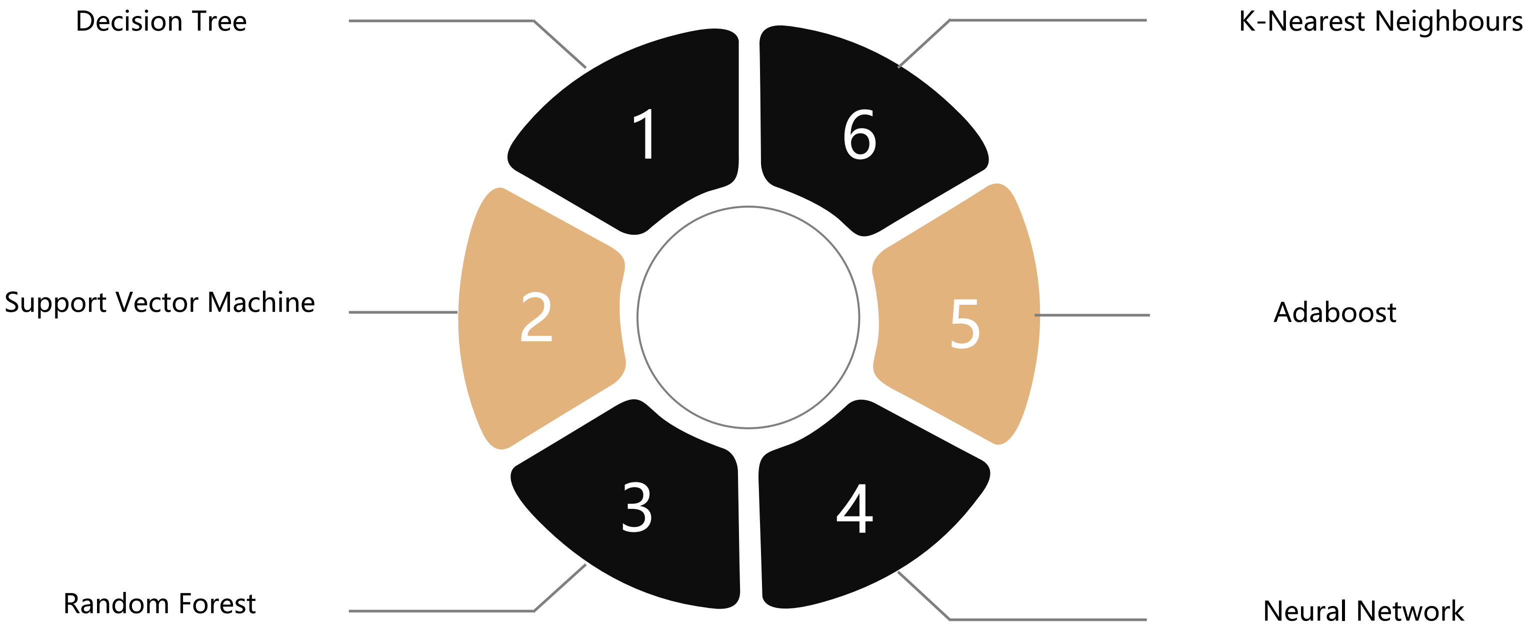
Feature	Score
exercised_stock_options	25.10
total_stock_value	24.47
bonus	21.06
salary	18.58
poi_email_ratio	16.24
deferred_income	11.60
long_term_incentive	10.07
restricted_stock	9.35
total_payments	8.87
shared_receipt_with_poi	8.75
loan_advances	7.24
expenses	6.23
from_poi_to_this_person	5.34
from_this_person_to_poi	2.43



04

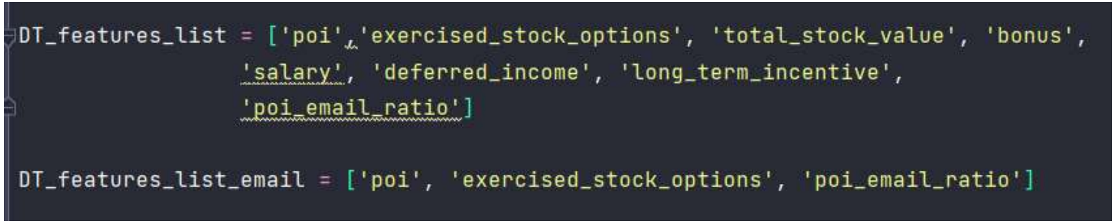
Determine Algorithm

Machine Learning Algorithm

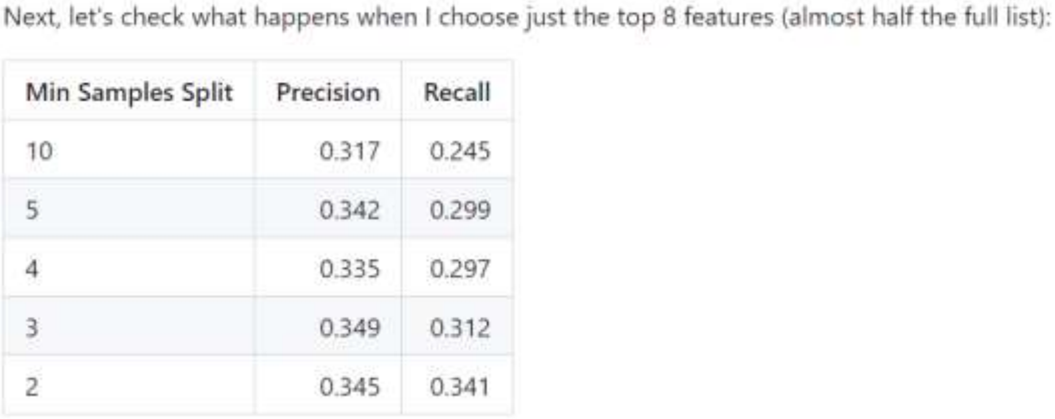


Select Features for ML Algorithm Process

After comparing with all the feature pairs, we decided to try just 3 features.
 The best financial feature, 'exercised_stock_options', and the best email feature, 'poi_email_ratio'.   a feature list of ['poi', 'exercised_stock_options', 'poi_email_ratio'].



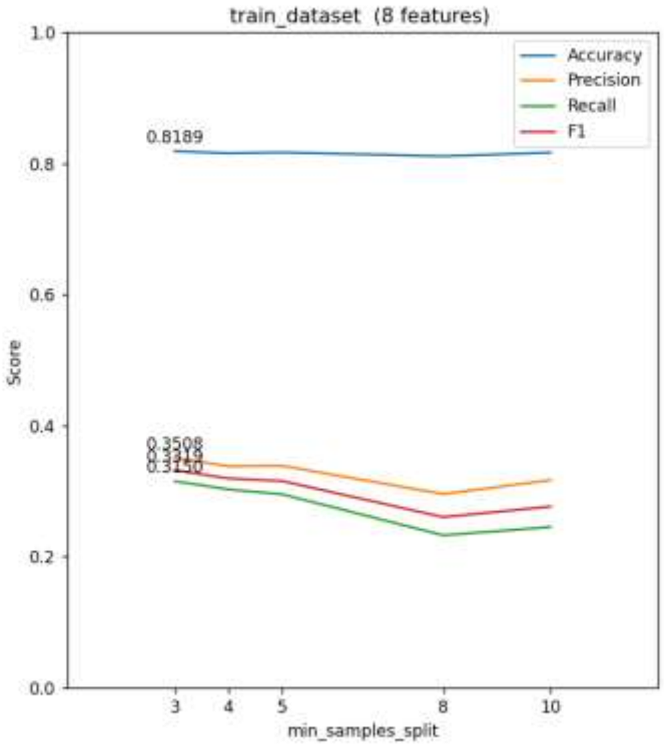
```
DT_features_list = ['poi','exercised_stock_options', 'total_stock_value', 'bonus',
                 'salary', 'deferred_income', 'long_term_incentive',
                 'poi_email_ratio']

DT_features_list_email = ['poi', 'exercised_stock_options', 'poi_email_ratio']
```

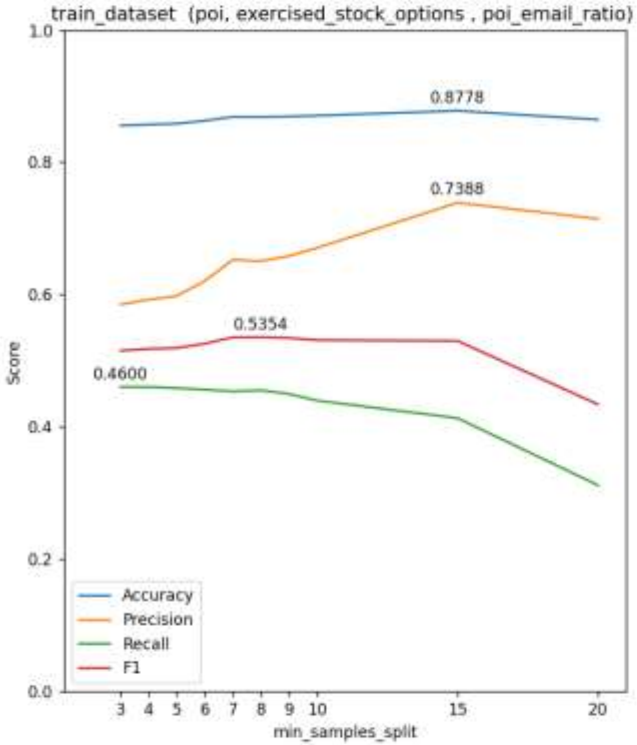


Next, let's check what happens when I choose just the top 8 features (almost half the full list):

Min Samples Split	Precision	Recall
10	0.317	0.245
5	0.342	0.299
4	0.335	0.297
3	0.349	0.312
2	0.345	0.341

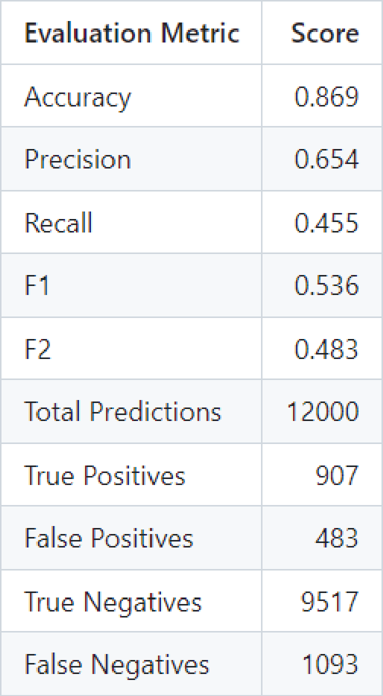
Select Features for ML Algorithm Process



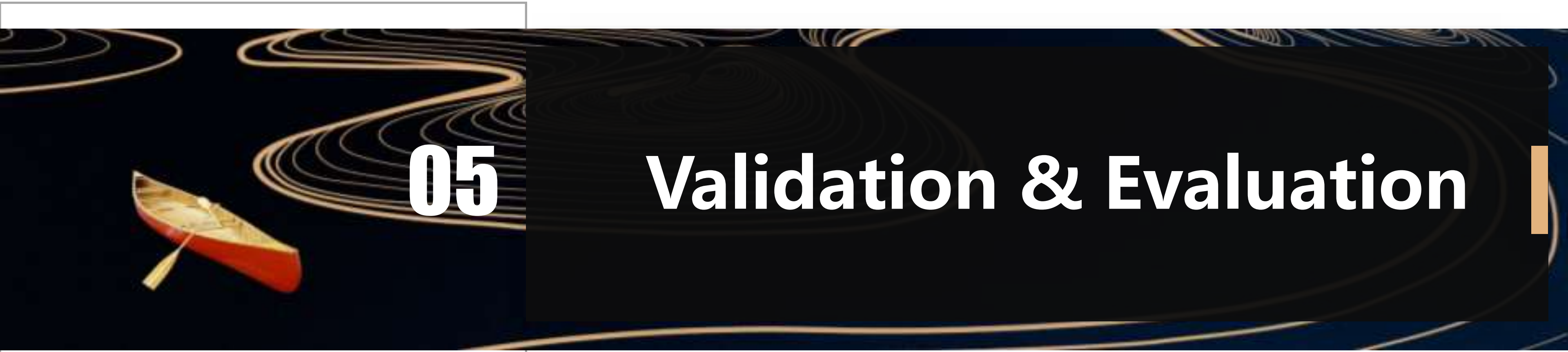


Select ML Algorithm Model

After calculation and comparison of various ml algorithms, the algorithm We've chosen is a **Decision Tree Classifier** with 'min_samples_split' value of 8 and a feature list of ['poi', 'exercised_stock_options', 'poi_email_ratio'].



Evaluation Metric	Score
Accuracy	0.869
Precision	0.654
Recall	0.455
F1	0.536
F2	0.483
Total Predictions	12000
True Positives	907
False Positives	483
True Negatives	9517
False Negatives	1093

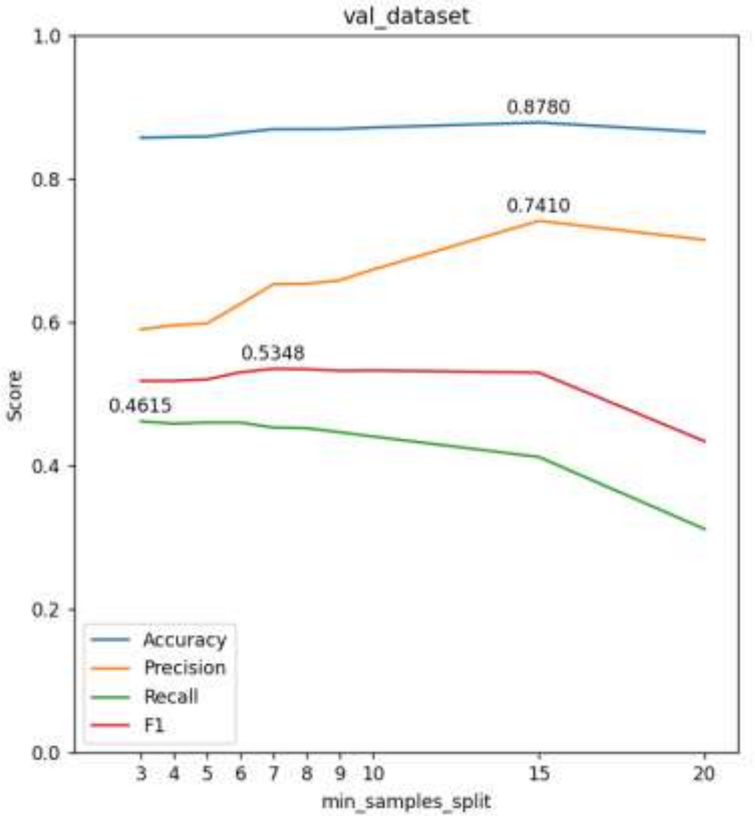


05 Validation & Evaluation

Validation & Evaluation

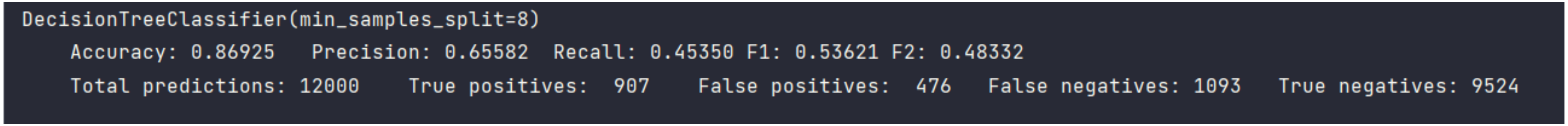
Use StratifiedShuffleSplit for the validation set.

At min_samples_split = 8, one of the reasons we chose min_samples_split was that F1 had the highest score.
It is F1 that takes into account both the accuracy rate and the recall rate, so that the performance of the classification model can be evaluated more comprehensively.
F1 = 2 * (Precision * Recall) / (Precision + Recall)



06 Conclusion

Conclusion



```
DecisionTreeClassifier(min_samples_split=8)
	Accuracy: 0.86925	Precision: 0.65582	Recall: 0.45350	F1: 0.53621	F2: 0.48332
	Total predictions: 12000	True positives:  907	False positives:  476	False negatives: 1093	True negatives: 9524
```

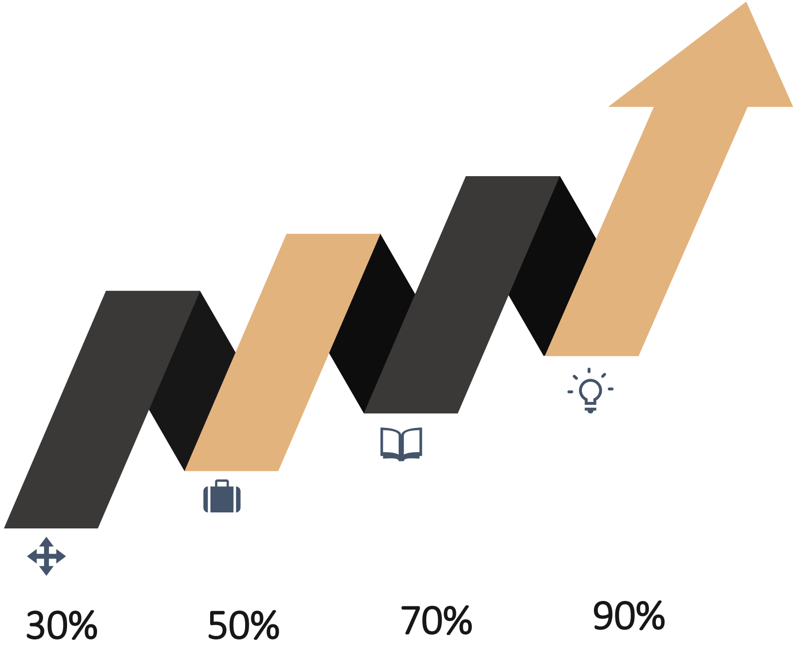
Based on the financial and email data combing with our ML model, we identified Enron employees who might have committed fraud.

Future Work

1. Tune Parameters

2. More ML Models

3. Innovation of Specific Model



Thanks for Watching