**1.**

**Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it. As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle those?**

**Background:**

*“Enron Corporation was an American energy, commodities, and services company based in Houston, Texas..*
*At the end of 2001, it was revealed that its reported financial condition was sustained by institutionalized, systematic, and creatively planned accounting fraud, known since as the Enron scandal. Enron has since become a well-known example of willful corporate fraud and corruption.”*
https://en.wikipedia.org/wiki/Enron

*“The collapse of Enron and subsequent public release of Enron data by the FERC has resulted in one of the largest and richest publicly available data sets for email research.”*
Enrondahttps://enrondata.readthedocs.io/en/latest/introduction/

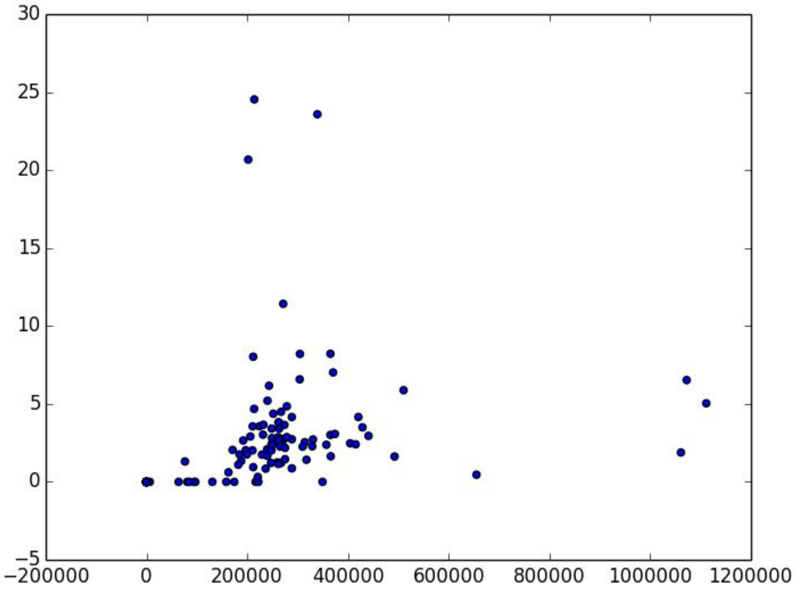
The objective for this project is to see if we can identify the culprits behind the Enron collapse using machine learning. The dataset contains financial and email features with POI labels for each person in the set.

**Data and outliers:**

There are 146 datapoints in the set. The task is to identify potential criminal behavior. Criminal behavior is by definition an outlier in modern society. So by simply looking for and removing outliers the chances of removing an important datapoint is significant. Therefore one should take care when attempting to remove any outlier as these are possibly the persons/datapoints that we’re after. This dataset contains 18 datapoints labeled as a person of interest and 128 non-poi’s.

From running trough the names in the dataset we can immediately identify two datapoints that should be removed: “TOTAL and THE TRAVEL AGENCY IN THE PARK”. Total is a summarization of the datapoints in the set and The travel agency in the park doesn’t identify a person (some further digging through registered companies in the region might reveal the persons behind the company but that is outside the scope of this exploration).

A feature called bonus_salary_ratio was created. It simply divides a person's bonus by their salary. For easier identification of possible outliers and possible poi’s. For visual identification the feature was plotted on a scatter chart against salaries for the datapoints.



From the plot we can see that one person(datapoint) received 25 times their salary as a bonus. Tracing the datapoint back to a person reveals that this point belongs to a BELDEN TIMOTHY N which is labeled as a person of interest. This probably indicates that this features isn’t useful for removing outliers, but the might be good indicator of a poi for the model.

**2.**
**What features did you end up using in your POI identifier, and what selection process did you use to pick them? Did you have to do any scaling? Why or why not? As part of the assignment, you should attempt to engineer your own feature that doesn’t come ready-made in the dataset–explain what feature you tried to make, and the rationale behind it. (You do not necessarily have to use it in the final analysis, only engineer and test it.) If you used an algorithm like a decision tree, please also give the feature importances of the features that you use.**

**Feature creation:**
Two features were added to the dataset.

- bonus_salary_ratio. As explained in above is the ratio between an individual's salary and bonus.
- Total_poi_email. The total_poi_email is just the to and from poi emails added together.

Both features was added to the final model.

**Scaling:**
For scaling MinMaxScaler was used. Normalizing every feature from 0 to 1 for consistency between features available for feature selection using SelctKBest. When using features with a wide range of values such as bonus and other features with a relatively tight range of values scaling is preferential.

“*If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.”*
https://en.wikipedia.org/wiki/Feature_scaling

**Feature selection:**

| Feature | Score | P value | Selected |
|---|---|---|---|
| salary | 18.289684 | 0.000035 | TRUE |
| bonus | 20.792252 | 0.000011 | TRUE |
| restricted_stock | 9.212811 | 0.002863 | TRUE |
| total_payments | 8.772778 | 0.003589 | TRUE |
| total_stock_value | 24.182899 | 0.000002 | TRUE |
| exercised_stock_options | 24.81508 | 0.000002 | TRUE |
| from_this_person_to_poi | 2.382612 | 0.124934 | TRUE |
| from_poi_to_this_person | 5.24345 | 0.023514 | TRUE |
| long_term_incentive | 9.922186 | 0.001994 | TRUE |
| director_fees | 2.126328 | 0.147011 | FALSE |
| expenses | 6.094173 | 0.014758 | TRUE |
| shared_receipt_with_poi | 8.589421 | 0.003946 | TRUE |
| to_messages | 1.646341 | 0.201563 | FALSE |
| from_messages | 0.169701 | 0.681003 | FALSE |
| loan_advances | 7.184056 | 0.008232 | TRUE |
| restricted_stock_deferred | 0.0655 | 0.798379 | FALSE |
| bonus_salary_ratio | 10.783585 | 0.001292 | TRUE |
| total_poi_emails | 4.863682 | 0.029047 | TRUE |

The models where “feed” 18 features consisting of financial and email information. In the final model only 14 features were selected. The reason for such a selection of features to add is probably because many of the different features explains the same underlying effects and reasons and therefore introduce multicollinearity in the model. PCA in turn flattened those features into two separate components.

***3.***

**What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms? [relevant rubric item: “pick an algorithm”**

**Model Selection process:**
This being a classification problem, three different models (GNB, SVC and Adaboost classifier) were inserted into a pipeline and GridSearchCV with a selection of parameters related to the respective models for tuning. of the Parameters the first step of the pipeline normalized the features as described above

| Model | Number of features | Components | Kernel | Whiten | C | Gamma | Estimators | Learning rate | Algorithm |
|---|---|---|---|---|---|---|---|---|---|
| SVC | 14 | 2 | linear | FALSE | 1 | auto | | | |
| Adaboost | 12 | 6 | | FALSE | | | 70 | 1.5 | SAMME.R |
| GNB | 4 | | | | | | | | |

After some run throughs with further manual selection and auto tuning SVC performed best with an F1 score of 0.45739:

### Model performance:

| Model | Accuracy | Precision | Recall | F1 | F2 |
|---|---|---|---|---|---|
| SVC | 0.71260 | 0.30563 | 0.90850 | 0.45739 | 0.65149 |
| Adaboost | 0.81667 | 0.23994 | 0.17300 | 0.20105 | 0.18322 |
| GNB | 0.84693 | 0.40842 | 0.33000 | 0.36504 | 0.34318 |

**4.**
**What does it mean to tune the parameters of an algorithm, and what can happen if you don’t do this well? How did you tune the parameters of your particular algorithm? (Some algorithms do not have parameters that you need to tune -- if this is the case for the one you picked, identify and briefly explain how you would have done it for the model that was not your final choice or a different model that does utilize parameter tuning, e.g. a decision tree classifier). [relevant rubric item: “tune the algorithm”]**

### Model performance:

Each algorithm requires parameters specific for it’s purpose and objective. In a classification algorithm tuning an algorithm can be described as setting the borders of where a class should be defined. Within what area should a point be defined as belonging to a class.

A classic issue is overtuning an algorithm. This is when the model performes perfect or near perfect on a set of training data, but handles unseen data poorly. This is generally a result of the model being to specific and less general the needed.

Below I’ve copied in the models used for this project from above:

| Model | Number of features | Components | Kernel | Whiten | C | Gamma | Estimators | Learning rate | Algorithm |
|---|---|---|---|---|---|---|---|---|---|
| SVC | 14 | 2 | linear | FALSE | 1 | auto | | | |
| Adaboost | 12 | 6 | | FALSE | | | 70 | 1.5 | SAMME.R |
| GNB | 4 | | | | | | | | |

**5.**
**What is validation, and what’s a classic mistake you can make if you do it wrong? How did you validate your analysis? [relevant rubric item: “validation strategy”]**

Validation is the process of excluding a portion of data from the training data for testing purposes on previously unseen data.

For this project StratifiedShuffleSplit was employed. It splits up the data with one portion for training purposes and holds out the remainging portion for testing purposes. Additionally StratifiedShuffleSplit can shuffles the data X amount (1000 times for this project) of times before the splitting takes place. This technique is especially useful to employ on small datasets (such as this project).

**6.**
**Give at least 2 evaluation metrics and your average performance for each of them. Explain an interpretation of your metrics that says something human-understandable about your algorithm’s performance. [relevant rubric item: “usage of evaluation metrics”]**

**Recall:**
Recall is the term used to describing how many of the True positives was actually found.
*Recall = True Positive / ( True Positive + False Negative)*
https://en.wikipedia.org/wiki/Precision_and_recall

**Precision:**
Precision is a term used to describe how many of the True positives was labeled as True positive by the model
*Precision = True Positive / ( True Positive + False Positive)*
https://en.wikipedia.org/wiki/Precision_and_recall