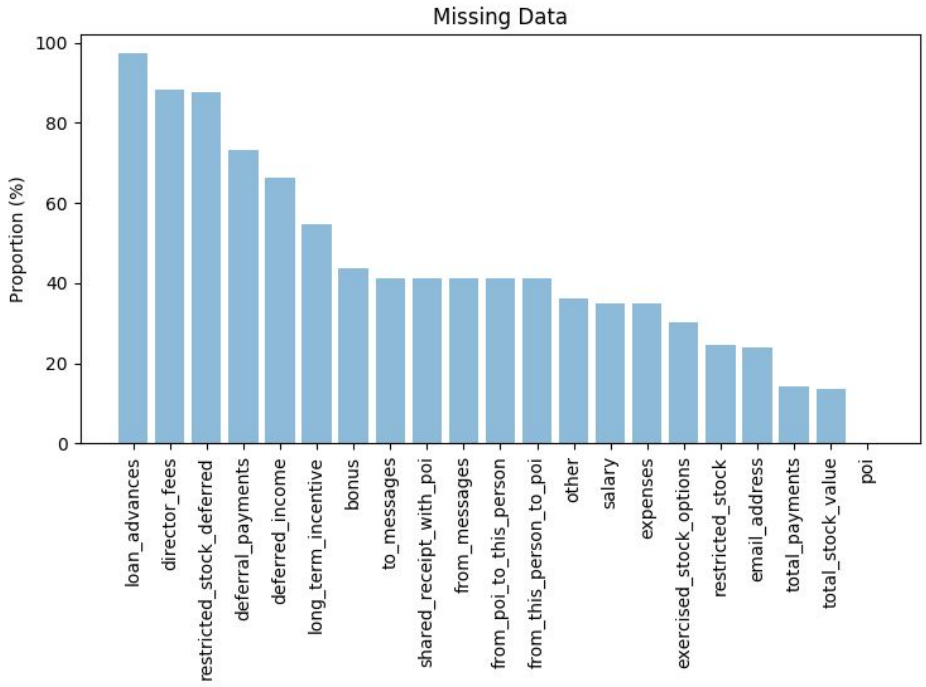
Enron Investigation

Jessica Matsuoka

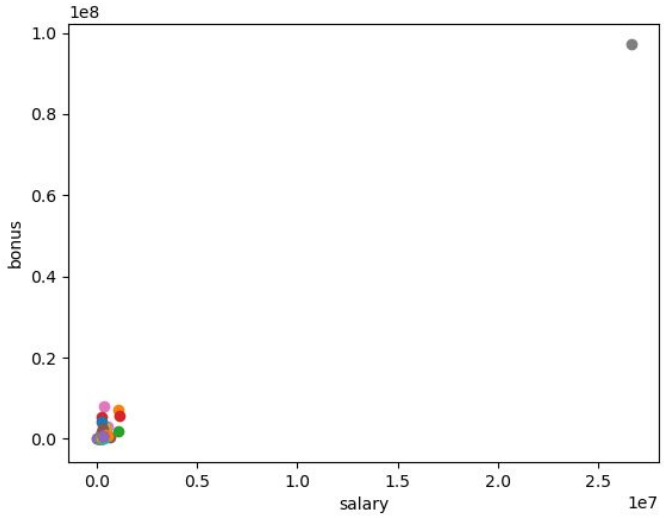
1. **Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it. As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle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

The main goal of the project is to explore a dataset with financial and email data from the Enron company, identifying frauds. Machine learning can help on this matter, by applying its techniques to develop a classifier to identify the people who were involved with the fraud.

The dataset is made up of employees' financial attributes and e-mails exchanged between them, plus an indicator if the person was involved in the fraud. The presented problem has defined classes (if the person is involved or not), and we can apply a supervised classifier to find patterns in the data and to classify the people according to the identified standards. The dataset consists of 146 observations and 21 characteristics. From the 146 observations, 18 are classified as POI's (person of interest). It was also observed that the dataset has missing data, in which loan_advances is the highest missing data variable with 97.26%.



Now, analyzing the outliers, we could find this data point when we were analyzing the variables salary and bonus. The extreme data point in the right, was a column names “Total”, which is the total Salary and Bonus. This data was removed from our dataset.



After removing this outlier, we found more 4 outliers. Investigating those data points, we found that 2 of them were POIs (SKILLING JEFFREY and LAY KENNETH). They were managers of Enron at that time.

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 does not 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.) In your feature selection step, if you used an algorithm like a decision tree, please also give the feature importances of the features that you use, and if you used an automated feature selection function like SelectKBest, please report the feature scores and reasons for your choice of parameter values. [relevant rubric items: “create new features”, “intelligently select features”, “properly scale features”]**

I tried 2 approaches to pick the features. The first approach is randomly selecting features that I thought could make difference in classification. This choice might have take time and luck. My first attempt was not successful.

The second approach was using the function SelectKBest from Sklearn Python LIbrary. In order to select the characteristics to be used, the k parameter iterated between

2 and 9 to compare the performance between the different values of k, 4-fold cross validation was used with the GaussianNB classifier for precision and recall.

Accuracy 0.733333333333
Precision = 0.5
Recall = 0.25

I added the feature poi_total. The purpose of this variable was to try to capture a communication pattern between the POIs, just like variables from_this_person_to _poi and from_poi_to _this _person. Assuming that the people involved in the fraud tend to communicate more and that in addition the criminal are divided between leaders and subordinates the following pattern would arise: POI's leaders who send many emails to subordinate POI's and subordinate POI's that receive many emails from leading POIs. If this hypothesis is true the POI's identified by the variable from _this_person _to_poi would be different from the POI's identified by the variable from_poi_to _this_person. With this in mind, the poi_total variable was created to try to capture the two patterns into a single variable. If we include the new variables into the Feature selection, the accuracy is higher.

Accuracy 0.8
Precision = 0.666666666667
Recall = 0.5

We want now to test which features will be tested with the classifiers:
::: PERFORMANCE WITH 2 FEATURES :::
exercised_stock_options
total_stock_value
Precision 0.379166666667
Recall 0.2875

::: PERFORMANCE WITH 3 FEATURES :::
exercised_stock_options
bonus
total_stock_value
Precision 0.3
Recall 0.4

::: PERFORMANCE WITH 4 FEATURES :::
salary
exercised_stock_options
bonus
total_stock_value

Precision 0.208333333333
Recall 0.225

::: PERFORMANCE WITH 5 FEATURES :::
salary
exercised_stock_options
bonus
total_stock_value
ratio_messages_to_poi
Precision 0.307142857143
Recall 0.425

::: PERFORMANCE WITH 6 FEATURES :::
salary
exercised_stock_options
bonus
total_stock_value
deferred_income
ratio_messages_to_poi
Precision 0.304861111111
Recall 0.375

::: PERFORMANCE WITH 7 FEATURES :::
salary
exercised_stock_options
bonus
total_stock_value
deferred_income
long_term_incentive
ratio_messages_to_poi
Precision 0.254166666667
Recall 0.325

::: PERFORMANCE WITH 8 FEATURES :::
salary
exercised_stock_options
bonus
restricted_stock
total_stock_value
deferred_income
long_term_incentive
ratio_messages_to_poi
Precision 0.316666666667

Recall 0.325

::: PERFORMANCE WITH 9 FEATURES :::
salary
exercised_stock_options
bonus
restricted_stock
total_stock_value
deferred_income
long_term_incentive
ratio_messages_to_poi
ratio_messages_shared
Precision 0.21130952381
Recall 0.2125

The performance with 5 features was chosen, due to Precision and Recall Score.

The features were scaled to range from 0 to 1, because one of the classifiers tested in the project was the Support Vector Machine, as it makes use of the Euclidean distance in its calculations, it was necessary to use variables of the same order of magnitude, in order to improve the efficiency of the classifier.

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"]**

The algorithms tested were Gaussian Naive Bayes, Decision Tree, AdaBoost and Support Vector Machine

Algorithm	Precision	Recall
GaussianNB	0.517	0.35
Decision Tree	0.405	0.438
Ada Boost	0.438	0.388
SVC	0	0

In terms of precision, GaussianNB performed better, which was the algorithm that I ended up using. I chose Precision over recall, because I took into consideration that is more important to have less False Positives in trade off to have more False Negatives. We don't want to involve innocent people into this event (Enron Fraud).

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? What parameters did you tune? (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 items: "discuss parameter tuning", "tune the algorithm"]**

Tuning the parameters is the process which optimizes the parameters that impact the model in order to enable the algorithm to perform the best (once, of course you have defined what "best" actual is). If the tuning step is poorly done two problems may arise: overfitting and underfitting. In the case of overfitting, the algorithm models the training data in a very specialized way and loses the ability to generalize to unseen data. While underfitting the algorithm does not assimilate the most significant trends of the training data. Both problems lead to poor algorithm performance.

In this, I used GridSearchCV function to calibrate the classifiers. The parameters were:

Decision Tree
- Criterion: gini, entropy
- Min sample split: 2, 3, 4
- Max depth: 1, 2, 3, 4
- Min samples leaf: 1, 2, 3, 4
- Max leaf node: 2, 3, 4, 5, 6

AdaBoost
- n_estimator: 20, 50, 70, 90
- Algorithm: SAMME, SAMME.R

Support Vector Machine
- Kernel: Rbf
  - gamma: [10^-10, 10^-9, ..., 10^10], C: [10^-10, 10^-9, ..., 10^10]
- Kernel: Sigmoid
  - gamma: [10^-20, 10^-19, ..., 10^20], coef0: [-40, -30, ..., 40]

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 items: "discuss validation", "validation strategy"]**

Validation is a step in the machine learning process used for the purpose of evaluating the real performance of the model. One of the techniques that can be used in validation is the division of data between two distinct sets: training and testing. The training set is used to train algorithm and the test set is used to evaluate the performance of the model. A classic error that can be identified in the validation is overfitting. It occurs when the model becomes extremely specific to the data used in training and loses the ability to generalize to previously unseen data. In the project, the GridSearchCV function was used, which besides performing parameter calibration uses cross-validation to verify the performance of each combination of parameters.

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"]**

Metric	**Value**
Precision	0.49
Recall	0.32

Performance is measured using 4-fold cross validation. The interpretation of these metrics is that precision is the number of items classified as POI by the model that are relevant, that is, of all people selected as POI by the model, about 49% are actually POI's.
While Recall is the number of relevant items classified as POI by the model, that is, of all POIs in the data set, about 32% were selected.