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 goal of this project is to identify a given person is a person of interest or not based information on each person. The data set contains the financial and email information of each person. Given data is used to train multiple classifiers and test the performance of each of those on test datasets.

After examining the dataset, by plotting, there was one outlier found which is a record for total of all data points. The data point was removed from the dataset before further processing.

Dataset structure:

No. of training examples (with outlier) – 146
No. of training examples (outlier removed) – 145

No. of POI data points – 18

No. of Non-POI data points – **127**

Dataset structure of Email training:

No. of POI emails – **18**

No of Non-POI emails – **18** (random sample on 127emails)

Total # of training examples – **7634**

Total no. of features pre-PCA – **32930**

Total no. of features post-PCA – **1000**

No. of training examples – **6107**

No. of test examples – **1527**

Oversampling + Under sampling:

Since the dataset was heavily imbalanced (18 POI and 127 non-poi) a combination of oversampling and under sampling was performed on the dataset to make it less biased towards the majority class.

Non-POI data points were under sampled from 127 to 70 by choosing random data points.

Poi data points were oversampled from 18 to 30 by choosing 3 sets of 10 random data points

So the composition of the dataset after the sampling process was as below

No. of training examples – 100

No. of POI data points – 30

No. of non-POI data points – 70

No. of Final features used - 11

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”, “properly scale features”, “intelligently select feature”]

Features Used:

Initial features used in the training

['salary','total_payments','from_this_person_to_poi','from_poi_to_this_person','shared_receipt_with_poi','bonus','total_stock_value','from_poi_frac','to_poi_frac']

Features used are have mix of email and finance data of each employee. Most important features are salary, from_poi_frac, to_poi_frac.

New Features:

New features created – **from_poi_frac, to_poi_frac**

These two new features give our model an ability to do more accurate classification by giving an emphasis on the interactions of a person with other POI among their overall interactions with everyone else.

from_poi_frac -> fraction of emails received from poi to total emails received.

to_poi_frac -> fraction of emails sent to poi to total emails sent.

Feature Scaling:

There was a minmax scaler applied to scale all features to be in the range of 0-1, since the range of feature values had very high difference between them, like Salary and from_poi_frac for instance.

Feature Importance (using K-Best):

Below are the feature importance scores after applying K best on all features (including the 2 new features)

Features	Importance Scores
to_poi_frac	18.6858803794
exercised_stock_options	18.4023914899
total_stock_value	17.9273198907
salary	16.7032447043
bonus	16.1529574277
long_term_incentive	14.1622827876
shared_receipt_with_poi	9.98054425897
expenses	9.5861841973
from_poi_to_this_person	9.49177860329
restricted_stock	8.1345379587
total_payments	7.81826293277
loan_advances	7.56722717425
other	6.64262888627
from_poi_frac	4.67232891469
deferred_income	4.03378144945
from_this_person_to_poi	3.70298216191
director_fees	3.28984674157
to_messages	2.78685757514
deferral_payments	0.809278496384
from_messages	0.299835369566
restricted_stock_deferred	0.274537872908

Feature Selection (GridSearch):

After identifying the importance of each features, multiple combinations of features (High scored features + few medium scored features + few low score features) were selected and tested.

The fits and test was first performed on 100 folds of split data.

Feature set -- different combinations of High scored + randomly selected few medium scores + randomly selected few low scores

Algorithm -- SVM with below parameter grids,

```
  clf_params= {'model__C': [60000,80000,100000],
      'model__kernel': ['rbf'],
      'model__tol': [1e-2,1e-3,1e-4,1e-1,0.5,0.7],
       'model__max_iter' : [1000,2000,3500,5000,10000,15000,25000,50000,60000]}
```

Training & Test split – Training – 0.90 and Test – 0.10

No. Shuffle split fold – 100

Below are some good metrics for the best estimator on different feature combinations,

Feature List	**Model params**	**Precision***	**Recall***
Feature_list_1	C=100000, iter=15000 tolerance = 0.001	0.735465116279	0.843333333333
Feature_list_2	C=60000, iter=15000, Tolerance = 0.01	0.690763052209	0.86
Feature_list_3	C=100000, iter=10000, Tolerance=0.7	0.709318497914	0.85
Feature_list_4	C=100000, iter=10000, Tolerance = 0.7	0.740997229917	0.891666666667
Feature_list_5	C=80000, iter=5000, Tolerance = 0.5	0.713888888889	0.856666666667
Feature_list_6	C=60000, iter=3500, Tolerance = 0.7	0.758875739645	0.855
Feature_list_7	C=80000, iter=10000, Tolerance = 0.01	0.697305863708	0.733333333333
Feature_list_8	C=60000,iter=2000, Tolerance = 0.5	0.734870317003	0.85

SVM GridSearch with 1000 Folds:

Based on the selected features and specific high performing parameters another round of fit and testing was performed on 1000 folds of the split data,

Filtered parameter set,

```
  clf_params_1000= {'model__C': [60000,80000,100000],
      'model__kernel': ['rbf'],
      'model__tol': [1e-2,1e-3,0.5,0.7],
       'model__max_iter' : [3500,5000,10000,15000]}
```

Below metrics and parameters were chosen after further fit and test on above selected features,

Feature List were chosen which had more balance between precision and recall.

Feature List	**Model Params**	**Precision**	**Recall**
Feature_list_1	C=100000,iter=10000, Tolerance = 0.5	0.715098094967	0.838333333333
Feature_list_4	C=100000,iter=15000, Tolerance = 0.01	0.744804655029	0.896
Feature_list_6	C=60000,iter=3500 Tolerance = 0.7	0.737403928266	0.8635

Feature_list_7	C=100000,iter=5000, Tolerance = 0.5	0.6751695665	0.763166666667

Decision Tree Classifier:

Based on the selected features list, a **Decision Tree** was applied on above features

Parameters

```
clf_params_dt = {'model__min_samples_split':[2,4,6,10]}
```

Feature List	Model params	Precision	Recall
Feature_list_1	Min_Split = 2	0.644107050952	0.834333333333
Feature_list_4	Min_Split = 2	0.725137150091	0.859166666667
Feature_list_6	Min_Split = 2	0.658562367865	0.830666666667
Feature_list_7	Min_Split = 2	0.72567114094	0.865

Based on the fit and tests run on different feature lists, below features were picked to form the final feature list and parameters,

Final run of fit/test was performed on above features and parameters

Final feature,

```
['poi','other','bonus','salary','restricted_stock','loan_advances','total_payme
nts','to_poi_frac','deferral_payments','from_poi_frac','shared_receipt_with_poi
','long_term_incentive']
```

Final parameters – SVM

```
{'model__C': 60000, 'model__max_iter': 3500, 'model__kernel': 'rbf',
'model__tol': 0.7}
```

Final parameters – Decision Tree

```
{'model__min_samples_split': 4}
```

Model	Model params	Precision	Recall
Gaussian SVM	C=60000,iter=3500, Tolerance = 0.7	0.733089021225	0.800166666667
Decision Tree	Min_Split = 4	0.699665831245	0.8375

SVM model has more balanced scores, so SVMs with below features and parameters were chosen for the final fit,

Final feature and parameters

```
['poi','other','bonus','salary','restricted_stock','loan_advances','total_payme
```

nts','to_poi_frac','deferral_payments','from_poi_frac','shared_receipt_with_poi','long_term_incentive']

```
{'model__C': 60000, 'model__max_iter': 3500, 'model__kernel': 'rbf',
'model__tol': 0.7}
```

Effects of the new features:

Below tables show the effect of the newly created the features,

['poi','other','bonus','salary','restricted_stock','loan_advances','total_payments','from_this_person_to_poi','deferral_payments','from_poi_to_this_person','shared_receipt_with_poi','long_term_incentive']

Classifier	Precision	Recall
SVM	0.690015986049	0.791333333333
Decision Tree	0.633354153654	0.845

Replacing the from and to email features with the fraction features,

['poi','other','bonus','salary','restricted_stock','loan_advances','total_payments','deferral_payments','from_poi_frac','to_poi_frac','shared_receipt_with_poi','long_term_incentive']

Classifier	Precision	Recall
SVM	0.733089021225	0.800166666667
Decision Tree	0.701646664806	0.838

The Recall stayed about the same but the precision seemed to get better with the new fraction features.

*Explanation for these parameters are explained in more detailed in question# 6. Short version – Higher and balanced values are better.

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”]

Algorithms used – Linear SVM, Gaussian SVM, Decision Tree, Gaussian Naïve Bayes.

Based on performance of different algorithms, Gaussian SVM performed best on the shuffled data.

Although when the Gaussian SVM was applied on the dataset without the shuffle split it did not provide good results.

Decision tree worked best when the classifier was applied on the split of dataset without the shuffle split.

Email processing and training:

Additionally, emails of all employees has been processed using text learning, vectorization and a Gaussian SVM has been used to train and evaluate metrics.

PCA has been applied and the dimension of features were reduced to 1000, since it had lot of features to process.

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"]

Parameter tuning means modifying the way the data is trained by an algorithm. By modifying the parameters, the fit can be made high biased or high variance. A successful parameter tuning aims to find a balance between the bias and variance.

A model with high recall is a high variance – low bias model (Does not generalize well for untrained data)

A model with high precision is a high bias and low variance model. (Generalizes well but miss placing data in the right class more often)

In this experiment I tried a higher min_samples_split(minimum samples needed to further proceed with the split) than the default value of 2, but it did not give best metrics than the default case

In the case of SVM I tried below parameters

Kernel = Linear, Gaussian

C values = [100000, 110000]

Iterations limit = [15000,20000,90000]

Best set of parameters has been chosen based on the test run on above different parameters.

In case of email processing to identify the POIs, PCA was used to tune the heavy number of features to a 1000.

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 checking the performance of a training algorithm on an independent dataset separate from the training dataset.

One big mistake on validation can be evaluating a training algorithm's performance using the same data it was trained on. Doing this will not only give a very optimistic metrics because it was specifically trained for the dataset, it also do not infer if an algorithm is over fitted/Very high variance.

Introducing the classifier to some new untrained data gives the true performance of a classifier and also detects overfitting and make sure the classifier generalizes well for untrained data.

In this experiment the dataset was split in to train and test dataset using a stratified shuffle split (since the training example was very limited)

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"]

Definition and reason to use Precision and Recall.

Precision – This is the measure of the algorithm's ability to accurately identify a POI among all the predictions it made for POI.

For example: In this POI identifier problem, the algorithm made a total of 15000 predictions (both POI (1) and Non-POI (0). Let's dissect more on how the algorithm performed in absolute terms,

Total Predictions = 20000

Out of those 20000, it predicted 6690 ones and 13310 Zeros.

Let's look at the actual ground truth setting aside the predictions.

# of actual POI (1) in the data set – 6000

# of actual Non-POI (0) in the data set – 14000

Now comparing the predictions with the ground truth, to see how many data points our algorithm correctly predicted,

It predicted 4865 data points accurately to be a poi, ie, both predicted and the ground truth are poi, which is called True positives.

True Positives = 4865

Same for Non-POI data points, algorithm accurately predicted 12175 data points to be non-poi. This is called True Negatives.

True Negatives = 12175

If we were to just calculate the accuracy of our algorithm, below proportion applies, Accuracy = (True positives + True Negatives)/Total Predictions

So the Accuracy of this dataset would be 83% and this is overly optimistic and a poor evaluation parameter. Let’s see why,

Let’s assume, for a min, that our algorithm just predicted 0 for all data points without doing any actual classification. Since the prediction are all zeros, our true negative will be same as the No. of actual zeros in our ground truth.

True Negatives = 14000
True Positives = 0

In this case our accuracy will be (**14000+0)/20000 = 0.7** and this is even without any actual classification. This is why accuracy is a bad metric for classification problems.

Enter the False Positives and False Negatives.

False Positives – Our algorithms predicted a data point as POI, but the ground truth was actually Non-POI.

False Negatives – Our algorithm predicted a data point as Non-POI, but the ground truth was actually POI.

Explaining the Precision and Recall in relation to above explanation,

Recall – The ability of our algorithm to label or find the poi data points.

Precision – The ability of our algorithm to label the poi data points accurately (both prediction and truth are same) among all the data points labeled as POI or recalled as POI.

Gaussian SVM with C = 60000, iteration = 3500

Recall: 0.72720

Algorithm was able to label a poi 72% percent of times.

Precision: 0.81083

Algorithm was able to label a poi accurately 81% of the times among all the data points it said was POI.

F1: 0.76675

A balanced score obtained by combining precision and recall and gives us a whole picture of our balance between precision and recall.

The model created using financial and email interaction did not perform so well on classifying the POIs without the under and over sampling.

So this below model is processing the email corpus to acquire a new dataset and train.

<u>Metrics on Email training algorithm</u>

Precision: 0.98

Recall: 0.97

F-Score 0.97

The algorithm trained on emails of poi and non-poi had excellent metrics, both precision and recall scores were excellent and it worked well on different test samples.

########### Feature Information for Question# 2 ###########

Feature Lists – 100 Folds

Feature_list_1 - ['poi', 'other', 'to_messages', 'bonus', 'from_this_person_to_poi', 'restricted_stock', 'from_poi_to_this_person','deferral_payments', 'loan_advances', 'salary', 'total_stock_value', 'total_payments']

Feature_list_2 - ['poi', 'exercised_stock_options', 'total_stock_value', 'bonus', 'salary', 'to_poi_frac', 'deferred_income','long_term_incentive', 'restricted_stock', 'total_payments', 'from_poi_to_this_person', 'loan_advances', 'restricted_stock_deferred']

Feature_list_3 - ['poi', 'exercised_stock_options', 'total_stock_value', 'long_term_incentive', 'to_poi_frac', 'loan_advances','shared_receipt_with_poi', 'from_poi_frac', 'director_fees', 'from_messages']

Feature_list_4 - ['poi', 'bonus', 'exercised_stock_options', 'to_poi_frac', 'restricted_stock', 'loan_advances','total_payments','other', 'director_fees', 'deferral_payments']

Feature_list_5 - ['poi', 'from_messages', 'total_payments', 'total_stock_value', 'to_poi_frac', 'deferral_payments','salary', 'exercised_stock_options', 'other', 'loan_advances', 'bonus', 'from_this_person_to_poi']

```
Feature_list_6 - ['poi', 'loan_advances', 'restricted_stock',
'long_term_incentive', 'restricted_stock_deferred','other', 'bonus',
'shared_receipt_with_poi', 'from_poi_frac',
'from_poi_to_this_person', 'salary','exercised_stock_options']

Feature_list_7 - ['poi', 'shared_receipt_with_poi', 'director_fees',
'other', 'deferral_payments', 'to_poi_frac','to_messages',
'total_payments', 'from_poi_to_this_person', 'expenses', 'salary',
'long_term_incentive']

Feature_list_8 - ['poi', 'to_poi_frac', 'exercised_stock_options',
'total_stock_value', 'salary', 'bonus','long_term_incentive',
'shared_receipt_with_poi', 'expenses', 'loan_advances',
'from_poi_frac', 'deferral_payments','restricted_stock_deferred']
```