**Introduction to Machine Learning @Udacity**

**Final Project: Identify suspects in Enron Fraud**

**1. Dataset and goal of project**

**Goal**

The main purpose of project is develop the machine learning algorithm to detect person of interest(POI) from dataset. A POI is someone who was indicted for fraud, settled with the government, or testified in exchange for immunity.

**Dataset**

We have Enron email+financial (E+F) dataset. It contains 146 Enron managers to investigate. Each sample in this dictionary is containing 21 features. 18 people from this dataset labeled as POI. There are two imbalanced classes (many more non-POIs than POIs). Here's an example of one POI data point:

LAY KENNETH L

salary : 1072321

to\_messages : 4273

deferral\_payments : 202911

total\_payments : 103559793

exercised\_stock\_options : 34348384

bonus : 7000000

restricted\_stock : 14761694

shared\_receipt\_with\_poi : 2411

restricted\_stock\_deferred: NaN

total\_stock\_value : 49110078

expenses : 99832

loan\_advances : 81525000

from\_messages : 36

other : 10359729

from\_this\_person\_to\_poi : 16

poi : 1

director\_fees : NaN

deferred\_income : -300000

long\_term\_incentive : 3600000

email\_address : kenneth.lay@enron.com

from\_poi\_to\_this\_person : 123

**Outliers**

Dataset contains some outliers. The TOTAL row is the biggest Enron E+F dataset outlier. We should remove it from dataset for reason it's a spreadsheet quirk. Moreover, there are 4 more outliers with big salary and bonus. Two people made bonuses more than 6 million dollars, and a salary of over 1 million dollars. There's no mistake in the data. Ken Lay and Jeffrey Skilling made such money. So, these data points should be left in and examine it with others.

**2. Feature selection process**

|  |  |
| --- | --- |
| Feature Selection | Justification |
| Expenses | PoI (100%) had expenses as compared to non-poi (60%) |
| Shared\_receipt\_with\_poi | (14/18 :77%) had shared receipt with PoI as compared to non-poi (77/127:57%) |
| From\_poi\_to\_this\_person | Emails from poi to the employee that might have material information |

**New features**

In addition I create two new features which were considered in course:

|  |  |
| --- | --- |
| Feature Selection | Justification |
| Total\_payments/Salary | total\_payments/salary to identify those who had most to gain from non-salaried compensation |
| Total\_stock\_value\_to\_payments | Total\_stock\_value/total\_payments to identify those who had most to gain from higher stock price |

Impact of New Features (using Decision Tree Classifier)

|  |  |
| --- | --- |
| Best scores without New Feature with DT | Best scores with New Features with DT |
| Accuracy: 0.82793  Precision: 0.36532  Recall: 0.39400  F1: 0.37912  F2: 0.38791 | Accuracy: 0.84443  Precision: 0.43697  Recall: 0.30850  F1: 0.36166  F2: 0.32777 |
| Total predictions: 15000  True positives: 788  False positives: 1369  False negatives: 1212  True negatives: 11631 | Total predictions: 14000  True positives: 617  False positives: 795  False negatives: 1383  True negatives: 11205 |

Feature selection process included several iterations. On the first step I created set of features based on data visualization and intuition. Then I examine seven classifiers on these features, and optimized them using pipeline with selectKBest, pca and MinMaxScaler. Of these, Decision Tree gave the best accuracy, precision and recall with following feature importance:

**Feature Importance:**

expenses 0.396

shared\_receipt\_with\_poi 0.271

payment\_to\_salary 0.070

total\_stock\_to\_payments 0.263

DecisionTreeClassifier(class\_weight=None, criterion='gini', max\_depth=4,

max\_features=None, max\_leaf\_nodes=None,

min\_impurity\_split=1e-07, min\_samples\_leaf=1,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

presort=False, random\_state=None, splitter='best')

Accuracy: 0.84443 Precision: 0.43697 Recall: 0.30850 F1: 0.36166 F2: 0.32777

Total predictions: 14000 True positives: 617 False positives: 795 False negatives: 1383 True negatives: 11205

Therefore, I chose the following features (dropping payment to salary) for the final run with following feature importance which improved precision to .53 and recall to .36! (see below for the full results)

**Selected Features:**

expenses

shared\_receipt\_with\_poi

total\_stock\_to\_payments

**3. Pick an algorithm**

**The following table describes all results of examination from the algorithm used:**

| **Algorithm** | **Pipeline** | **Accuracy** | **Precision** | **Recall** | **F1** | **F2** |
| --- | --- | --- | --- | --- | --- | --- |
| **Naive Bayes** | No | 0.67838 | 0. 19497 | 0. 34850 | 0. 25004 | 0. 30108 |
| Yes | 0. 72746 | 0. 21792 | 0. 29800 | 0. 25174 | 0. 27760 |
| **SVM** | No\* | Got a divide by zero when trying out: SVC(C=0.1, cache\_size=200, class\_weight=None, coef0=0.0, decision\_function\_shape=None, degree=3, gamma=0.1, kernel='rbf', max\_iter=-1, probability=False, random\_state=None, shrinking=True, tol=0.001, verbose=False) | | | | |
| Yes\* |
| **Decision Tree** | **No** | **0. 85292** | **0. 53235** | **0. 36200** | **0. 43095** | **0. 38675** |
| Yes | 0. 82408 | 0. 37951 | 0. 22600 | 0. 28330 | 0. 24589 |
| **Nearest Neighbors** | No | 0.78369 | 0.18232 | 0.11650 | 0.14216 | 0.12557 |
| Yes | 0.78046 | 0.19759 | 0.13950 | 0.16354 | 0.14822 |
| **Random Forest** | No | 0.77738 | 0.28468 | 0.29550 | 0.28999 | 0.29327 |
| Yes | 0.77785 | 0.28937 | 0.30500 | 0.29698 | 0.30174 |
| **AdaBoost** | No | 0.81585 | 0.33852 | 0.20650 | 0.25652 | 0.22397 |
| Yes | 0.78877 | 0.24347 | 0.17700 | 0.20498 | 0.18722 |
| **QDA** | No | 0.67600 | 0.18237 | 0.31750 | 0.23167 | 0.27652 |
| Yes | 0.67600 | : 0.18237 | 0.31750 | 0.23167 | 0.27652 |

**Chosen algorithm**

Based on best performance level I picked Decision Tree as a final algorithm.

**4. Tune the algorithm**

**Reasons for algorithm tuning**

The main reason is to get better results from algorithm. I used **GridSearchCV** with following parameters to tune the algorithm.

| **Parameter** | **Settings for**  **investigation** | **Best Value** |
| --- | --- | --- |
| min\_samples\_split | [2,4,6,8] | 2 |
| splitter | ['random','best'] | best |
| max\_depth | [2,4,6,8,10,15] | 4 |
| criterian | [‘gini’,’entropy’] | ‘entropy’ |

**5. Validation**

To validate my analysis I used [stratified shuffle split cross validation](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedShuffleSplit.html) developed by Udacity and defined in tester.py file. I had to modify test\_classifier to return all the computed metrics for comparison with prevailing values. In addition, the input arrays to fit function had to numpy arrays for the pipeline classifier.

**6. Evaluation metrics**

I used precision and recall evaluation metrics to estimate model. Final results can be found in table below

| **Accuracy** | **Precision** | **Recall** | **F1** | **F2** |
| --- | --- | --- | --- | --- |
| 0.88100 | 0.71228 | 0.38 | 0.49560 | 0.41910 |
| **True Positive** | **False Positive** | **False Negative** | **True Negatives** | **Total** |
| 760 | 307 | 1240 | 10693 | 13000 |

**Conclusion**

With Precision of .71 and Recall of .38, project goal of higher than .3 was reached. In this example, higher precision is more important as we want to minimize innocent employees identified as poi suspects. At the same time Recall = 0.38 says only 38% of all POIs were identified.

We have very imbalanced classes in E+F dataset. In addition, almost half of all POIs weren't included in dataset. Under the circumstances, the result received is quite good though it's not ideal, of course.

**Outputs log**

PS C:\Users\KUMAR-NUC\Documents\GitHub\machine-learning\final\_project> python .\poi\_id.py

C:\Users\KUMAR-NUC\Anaconda2\lib\site-packages\sklearn\cross\_validation.py:44: DeprecationWarning: This module was deprecated in version 0.18 in favor of the model\_selection module into which all the refactored classes an

d functions are moved. Also note that the interface of the new CV iterators are different from that of this module. This module will be removed in 0.20.

"This module will be removed in 0.20.", DeprecationWarning)

number of features: 4

number of keys: 145

number of samples: 145

emp name poi expenses shared\_receipt\_with\_poi total\_stock\_to\_payments

1 ALLEN PHILLIP K 0.0 13868.00 1407.00 0.39

2 BADUM JAMES P 0.0 3486.00 0.00 1.41

3 BANNANTINE JAMES M 0.0 56301.00 465.00 5.72

4 BAXTER JOHN C 0.0 11200.00 0.00 1.89

5 BAY FRANKLIN R 0.0 129142.00 0.00 0.08

6 BAZELIDES PHILIP J 0.0 0.00 0.00 1.86

7 BECK SALLY W 0.0 37172.00 2639.00 0.13

8 BELDEN TIMOTHY N 1.0 17355.00 5521.00 0.20

9 BELFER ROBERT 0.0 0.00 0.00 -0.43

10 BERBERIAN DAVID 0.0 11892.00 0.00 10.91

11 BERGSIEKER RICHARD P 0.0 59175.00 233.00 1.07

12 BHATNAGAR SANJAY 0.0 0.00 463.00 0.00

13 BIBI PHILIPPE A 0.0 38559.00 1336.00 0.90

14 BLACHMAN JEREMY M 0.0 84208.00 2326.00 0.47

15 BLAKE JR. NORMAN P 0.0 1279.00 0.00 0.00

16 BOWEN JR RAYMOND M 1.0 65907.00 1593.00 0.09

17 BROWN MICHAEL 0.0 49288.00 761.00 0.00

18 BUCHANAN HAROLD G 0.0 600.00 23.00 0.96

19 BUTTS ROBERT H 0.0 9410.00 0.00 0.33

20 BUY RICHARD B 0.0 0.00 2333.00 1.46

21 CALGER CHRISTOPHER F 1.0 35818.00 2188.00 0.08

22 CARTER REBECCA C 0.0 0.00 196.00 0.00

23 CAUSEY RICHARD A 1.0 30674.00 1585.00 1.34

24 CHAN RONNIE 0.0 0.00 0.00 0.00

25 CHRISTODOULOU DIOMED 0.0 0.00 0.00 0.00

26 CLINE KENNETH W 0.0 0.00 0.00 0.00

27 COLWELL WESLEY 1.0 16514.00 1132.00 0.47

28 CORDES WILLIAM R 0.0 0.00 58.00 0.00

29 COX DAVID 0.0 27861.00 71.00 0.45

30 CUMBERLAND MICHAEL S 0.0 22344.00 0.00 0.26

31 DEFFNER JOSEPH M 0.0 41626.00 552.00 0.13

32 DELAINEY DAVID W 1.0 86174.00 2097.00 0.76

33 DERRICK JR. JAMES V 0.0 51124.00 1401.00 16.03

34 DETMERING TIMOTHY J 0.0 52255.00 0.00 1.68

35 DIETRICH JANET R 0.0 3475.00 1902.00 1.32

36 DIMICHELE RICHARD G 0.0 35812.00 0.00 3.51

37 DODSON KEITH 0.0 28164.00 114.00 0.00

38 DONAHUE JR JEFFREY M 0.0 96268.00 772.00 1.23

39 DUNCAN JOHN H 0.0 0.00 0.00 4.80

40 DURAN WILLIAM D 0.0 25785.00 599.00 0.78

41 ECHOLS JOHN B 0.0 21530.00 0.00 0.37

42 ELLIOTT STEVEN 0.0 78552.00 0.00 31.54

43 FALLON JAMES B 0.0 95924.00 1604.00 0.63

44 FASTOW ANDREW S 1.0 55921.00 0.00 0.74

45 FITZGERALD JAY L 0.0 23870.00 723.00 1.15

46 FOWLER PEGGY 0.0 0.00 10.00 0.00

47 FOY JOE 0.0 0.00 2.00 1.89

48 FREVERT MARK A 0.0 86987.00 2979.00 0.85

49 FUGH JOHN L 0.0 0.00 0.00 3.49

50 GAHN ROBERT S 0.0 50080.00 0.00 0.35

51 GARLAND C KEVIN 0.0 48405.00 178.00 0.57

52 GATHMANN WILLIAM D 0.0 0.00 0.00 0.00

53 GIBBS DANA R 0.0 0.00 23.00 2.30

54 GILLIS JOHN 0.0 0.00 0.00 0.00

55 GLISAN JR BEN F 1.0 125978.00 874.00 0.61

56 GOLD JOSEPH 0.0 0.00 0.00 0.41

57 GRAMM WENDY L 0.0 0.00 0.00 0.00

58 GRAY RODNEY 0.0 0.00 0.00 0.00

59 HAEDICKE MARK E 0.0 76169.00 1847.00 0.21

60 HANNON KEVIN P 1.0 34039.00 1035.00 22.14

61 HAUG DAVID L 0.0 475.00 471.00 4668.00

62 HAYES ROBERT E 0.0 0.00 50.00 19.02

63 HAYSLETT RODERICK J 0.0 0.00 571.00 0.00

64 HERMANN ROBERT J 0.0 48357.00 0.00 0.51

65 HICKERSON GARY J 0.0 98849.00 900.00 0.21

66 HIRKO JOSEPH 1.0 77978.00 0.00 337.74

67 HORTON STANLEY C 0.0 0.00 1074.00 2.32

68 HUGHES JAMES A 0.0 0.00 589.00 0.00

69 HUMPHREY GENE E 0.0 4994.00 119.00 0.74

70 IZZO LAWRENCE L 0.0 28093.00 437.00 2.94

71 JACKSON CHARLENE R 0.0 10181.00 117.00 1.32

72 JAEDICKE ROBERT 0.0 0.00 0.00 5.16

73 KAMINSKI WINCENTY J 0.0 83585.00 583.00 0.90

74 KEAN STEVEN J 0.0 41953.00 3639.00 3.52

75 KISHKILL JOSEPH G 0.0 116335.00 0.00 1.47

76 KITCHEN LOUISE 0.0 5774.00 3669.00 0.16

77 KOENIG MARK E 1.0 127017.00 2271.00 1.21

78 KOPPER MICHAEL J 1.0 118134.00 0.00 0.37

79 LAVORATO JOHN J 0.0 49537.00 3962.00 0.50

80 LAY KENNETH L 1.0 99832.00 2411.00 0.47

81 LEFF DANIEL P 0.0 0.00 2672.00 0.14

82 LEMAISTRE CHARLES 0.0 0.00 0.00 4.72

83 LEWIS RICHARD 0.0 0.00 739.00 0.00

84 LINDHOLM TOD A 0.0 57727.00 0.00 3.50

85 LOCKHART EUGENE E 0.0 0.00 0.00 0.00

86 LOWRY CHARLES P 0.0 0.00 0.00 0.00

87 MARTIN AMANDA K 0.0 8211.00 477.00 0.25

88 MCCARTY DANNY J 0.0 0.00 508.00 0.00

89 MCCLELLAN GEORGE 0.0 228763.00 1469.00 0.72

90 MCCONNELL MICHAEL S 0.0 81364.00 2189.00 1.48

91 MCDONALD REBECCA 0.0 0.00 720.00 0.00

92 MCMAHON JEFFREY 0.0 137108.00 2228.00 0.41

93 MENDELSOHN JOHN 0.0 148.00 0.00 0.00

94 METTS MARK 0.0 94299.00 702.00 0.55

95 MEYER JEROME J 0.0 2151.00 0.00 0.00

96 MEYER ROCKFORD G 0.0 0.00 22.00 0.52

97 MORAN MICHAEL P 0.0 0.00 127.00 0.00

98 MORDAUNT KRISTINA M 0.0 35018.00 0.00 0.33

99 MULLER MARK S 0.0 0.00 114.00 0.44

100 MURRAY JULIA H 0.0 57580.00 395.00 0.74

101 NOLES JAMES L 0.0 0.00 0.00 0.48

102 OLSON CINDY K 0.0 63791.00 856.00 1.97

103 OVERDYKE JR JERE C 0.0 18834.00 0.00 29.26

104 PAI LOU L 0.0 32047.00 0.00 7.63

105 PEREIRA PAULO V. FER 0.0 27942.00 0.00 0.00

106 PICKERING MARK R 0.0 31653.00 728.00 0.02

107 PIPER GREGORY F 0.0 43057.00 742.00 0.51

108 PIRO JIM 0.0 0.00 3.00 0.00

109 POWERS WILLIAM 0.0 0.00 12.00 0.00

110 PRENTICE JAMES 0.0 0.00 0.00 1.94

111 REDMOND BRIAN L 0.0 14689.00 1063.00 70.75

112 REYNOLDS LAWRENCE 0.0 8409.00 0.00 10.70

113 RICE KENNETH D 1.0 46950.00 864.00 44.63

114 RIEKER PAULA H 1.0 33271.00 1258.00 1.75

115 SAVAGE FRANK 0.0 0.00 0.00 0.00

116 SCRIMSHAW MATTHEW 0.0 0.00 0.00 0.00

117 SHANKMAN JEFFREY A 0.0 178979.00 1730.00 0.68

118 SHAPIRO RICHARD S 0.0 137767.00 4527.00 0.93

119 SHARP VICTORIA T 0.0 116337.00 2477.00 0.31

120 SHELBY REX 1.0 22884.00 91.00 1.24

121 SHERRICK JEFFREY B 0.0 0.00 583.00 0.00

122 SHERRIFF JOHN R 0.0 0.00 2103.00 0.72

123 SKILLING JEFFREY K 1.0 29336.00 2042.00 3.01

124 STABLER FRANK 0.0 16514.00 0.00 0.46

125 SULLIVAN-SHAKLOVITZ 0.0 0.00 0.00 1.36

126 SUNDE MARTIN 0.0 0.00 2565.00 0.45

127 TAYLOR MITCHELL S 0.0 0.00 300.00 3.43

128 THE TRAVEL AGENCY IN 0.0 0.00 0.00 0.00

129 THORN TERENCE H 0.0 46145.00 73.00 5.29

130 TILNEY ELIZABETH A 0.0 0.00 379.00 2.92

131 UMANOFF ADAM S 0.0 53122.00 41.00 0.00

132 URQUHART JOHN A 0.0 228656.00 0.00 0.00

133 WAKEHAM JOHN 0.0 103773.00 0.00 0.00

134 WALLS JR ROBERT H 0.0 50936.00 215.00 3.28

135 WALTERS GARETH W 0.0 33785.00 0.00 11.79

136 WASAFF GEORGE 0.0 0.00 337.00 1.99

137 WESTFAHL RICHARD K 0.0 51870.00 0.00 0.51

138 WHALEY DAVID A 0.0 0.00 0.00 0.00

139 WHALLEY LAWRENCE G 0.0 57838.00 3920.00 1.30

140 WHITE JR THOMAS E 0.0 81353.00 0.00 7.83

141 WINOKUR JR. HERBERT 0.0 1413.00 0.00 0.00

142 WODRASKA JOHN 0.0 0.00 0.00 0.00

143 WROBEL BRUCE 0.0 0.00 0.00 0.00

144 YEAGER F SCOTT 1.0 53947.00 0.00 32.99

145 YEAP SOON 0.0 55097.00 0.00 3.50

Performing classification using Naive Bayes

GaussianNB(priors=None)

Accuracy: 0.67838 Precision: 0.19497 Recall: 0.34850 F1: 0.25004 F2: 0.30108

Total predictions: 13000 True positives: 697 False positives: 2878 False negatives: 1303 True negatives: 8122

Performing classification using SVM

Got a divide by zero when trying out: SVC(C=0.1, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape=None, degree=3, gamma=0.1, kernel='rbf',

max\_iter=-1, probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

Precision or recall may be undefined due to a lack of true positive predicitons.

Performing classification using Decision Tree

Feature Importance:

expenses 0.599

shared\_receipt\_with\_poi 0.164

total\_stock\_to\_payments 0.236

DecisionTreeClassifier(class\_weight=None, criterion='entropy', max\_depth=4,

max\_features=None, max\_leaf\_nodes=None,

min\_impurity\_split=1e-07, min\_samples\_leaf=1,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

presort=False, random\_state=None, splitter='best')

Accuracy: 0.88100 Precision: 0.71228 Recall: 0.38000 F1: 0.49560 F2: 0.41910

Total predictions: 13000 True positives: 760 False positives: 307 False negatives: 1240 True negatives: 10693

\*\*\*found better precision and recall: 0.712277413308 0.38

Performing classification using Nearest Neighbors

KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski',

metric\_params=None, n\_jobs=1, n\_neighbors=3, p=2,

weights='distance')

Accuracy: 0.78369 Precision: 0.18232 Recall: 0.11650 F1: 0.14216 F2: 0.12557

Total predictions: 13000 True positives: 233 False positives: 1045 False negatives: 1767 True negatives: 9955

Performing classification using Random Forest

RandomForestClassifier(bootstrap=True, class\_weight='balanced',

criterion='gini', max\_depth=None, max\_features='auto',

max\_leaf\_nodes=None, min\_impurity\_split=1e-07,

min\_samples\_leaf=3, min\_samples\_split=3,

min\_weight\_fraction\_leaf=0.0, n\_estimators=10, n\_jobs=1,

oob\_score=False, random\_state=42, verbose=0, warm\_start=False)

Accuracy: 0.77738 Precision: 0.28468 Recall: 0.29550 F1: 0.28999 F2: 0.29327

Total predictions: 13000 True positives: 591 False positives: 1485 False negatives: 1409 True negatives: 9515

Performing classification using AdaBoost

AdaBoostClassifier(algorithm='SAMME.R',

base\_estimator=DecisionTreeClassifier(class\_weight=None, criterion='gini', max\_depth=4,

max\_features=None, max\_leaf\_nodes=None,

min\_impurity\_split=1e-07, min\_samples\_leaf=1,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

presort=False, random\_state=None, splitter='best'),

learning\_rate=1.0, n\_estimators=50, random\_state=0)

Accuracy: 0.81585 Precision: 0.33852 Recall: 0.20650 F1: 0.25652 F2: 0.22397

Total predictions: 13000 True positives: 413 False positives: 807 False negatives: 1587 True negatives: 10193

Performing classification using QDA

QuadraticDiscriminantAnalysis(priors=None, reg\_param=0.0,

store\_covariances=False, tol=0.0001)

Accuracy: 0.67600 Precision: 0.18237 Recall: 0.31750 F1: 0.23167 F2: 0.27652

Total predictions: 13000 True positives: 635 False positives: 2847 False negatives: 1365 True negatives: 8153

Performing classification using Naive Bayes with pca

Pipeline(steps=[('feature\_selection', SelectKBest(k='all', score\_func=<function f\_classif at 0x0000000005705A58>)), ('pca', PCA(copy=True, iterated\_power='auto', n\_components=3, random\_state=None,

svd\_solver='randomized', tol=0.0, whiten=True)), ('clf', GaussianNB(priors=None))])

Accuracy: 0.72746 Precision: 0.21792 Recall: 0.29800 F1: 0.25174 F2: 0.27760

Total predictions: 13000 True positives: 596 False positives: 2139 False negatives: 1404 True negatives: 8861

Performing classification using SVM with pca

Got a divide by zero when trying out: Pipeline(steps=[('feature\_selection', SelectKBest(k='all', score\_func=<function f\_classif at 0x0000000005705A58>)), ('pca', PCA(copy=True, iterated\_power='auto', n\_components=3, rando

m\_state=None,

svd\_solver='randomized', tol=0.0, whiten=True)), ('clf', SVC(C=0.1, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape=None, degree=3, gamma=0.1, kernel='rbf',

max\_iter=-1, probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False))])

Precision or recall may be undefined due to a lack of true positive predicitons.

Performing classification using Decision Tree with pca

Pipeline(steps=[('feature\_selection', SelectKBest(k='all', score\_func=<function f\_classif at 0x0000000005705A58>)), ('pca', PCA(copy=True, iterated\_power='auto', n\_components=3, random\_state=None,

svd\_solver='randomized', tol=0.0, whiten=True)), ('clf', DecisionTreeClassifier(class\_weight=None, criterion='...it=2, min\_weight\_fraction\_leaf=0.0,

presort=False, random\_state=None, splitter='best'))])

Accuracy: 0.84769 Precision: 0.50977 Recall: 0.26100 F1: 0.34524 F2: 0.28923

Total predictions: 13000 True positives: 522 False positives: 502 False negatives: 1478 True negatives: 10498

Performing classification using Nearest Neighbors with pca

Pipeline(steps=[('feature\_selection', SelectKBest(k='all', score\_func=<function f\_classif at 0x0000000005705A58>)), ('pca', PCA(copy=True, iterated\_power='auto', n\_components=3, random\_state=None,

svd\_solver='randomized', tol=0.0, whiten=True)), ('clf', KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski',

metric\_params=None, n\_jobs=1, n\_neighbors=3, p=2,

weights='distance'))])

Accuracy: 0.78046 Precision: 0.19759 Recall: 0.13950 F1: 0.16354 F2: 0.14822

Total predictions: 13000 True positives: 279 False positives: 1133 False negatives: 1721 True negatives: 9867

Performing classification using Random Forest with pca

Pipeline(steps=[('feature\_selection', SelectKBest(k='all', score\_func=<function f\_classif at 0x0000000005705A58>)), ('pca', PCA(copy=True, iterated\_power='auto', n\_components=3, random\_state=None,

svd\_solver='randomized', tol=0.0, whiten=True)), ('clf', RandomForestClassifier(bootstrap=True, class\_weight='...stimators=10, n\_jobs=1,

oob\_score=False, random\_state=42, verbose=0, warm\_start=False))])

Accuracy: 0.77785 Precision: 0.28937 Recall: 0.30500 F1: 0.29698 F2: 0.30174

Total predictions: 13000 True positives: 610 False positives: 1498 False negatives: 1390 True negatives: 9502

Performing classification using AdaBoost with pca

Pipeline(steps=[('feature\_selection', SelectKBest(k='all', score\_func=<function f\_classif at 0x0000000005705A58>)), ('pca', PCA(copy=True, iterated\_power='auto', n\_components=3, random\_state=None,

svd\_solver='randomized', tol=0.0, whiten=True)), ('clf', AdaBoostClassifier(algorithm='SAMME.R',

bas...random\_state=None, splitter='best'),

learning\_rate=1.0, n\_estimators=50, random\_state=0))])

Accuracy: 0.78877 Precision: 0.24347 Recall: 0.17700 F1: 0.20498 F2: 0.18722

Total predictions: 13000 True positives: 354 False positives: 1100 False negatives: 1646 True negatives: 9900

Performing classification using QDA with pca

Pipeline(steps=[('feature\_selection', SelectKBest(k='all', score\_func=<function f\_classif at 0x0000000005705A58>)), ('pca', PCA(copy=True, iterated\_power='auto', n\_components=3, random\_state=None,

svd\_solver='randomized', tol=0.0, whiten=True)), ('clf', QuadraticDiscriminantAnalysis(priors=None, reg\_param=0.0,

store\_covariances=False, tol=0.0001))])

Accuracy: 0.67600 Precision: 0.18237 Recall: 0.31750 F1: 0.23167 F2: 0.27652

Total predictions: 13000 True positives: 635 False positives: 2847 False negatives: 1365 True negatives: 8153