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”]

   O objetivo do projeto é criar um modelo de machine learning capaz de indentificar as pessoas suspeitas envolvidas no caso de corrupção da Enron denomidadas como person of interest (POI), escolhendo e configurando algoritimos para obter o melhor resultado possivel. O cojuto de dados fornercidos são referentes aos dados financeiros e emails de 146 funcionários da Enron sendo sendo 18 POI e 128 não POI.

   No comjunto de dados,foi indentificados muitos valores faltantes (NaN) como mostra na tabela abaixo:

	Vars	NaN	P_Var_NaN
salary	95	51	0.349315
to_messages	86	60	0.410959
deferral_payments	39	107	0.732877
total_payments	125	21	0.143836
exercised_stock_options	102	44	0.301370
bonus	82	64	0.438356
restricted_stock	110	36	0.246575
shared_receipt_with_poi	86	60	0.410959
restricted_stock_deferred	18	128	0.876712
total_stock_value	126	20	0.136986
expenses	95	51	0.349315
loan_advances	4	142	0.972603

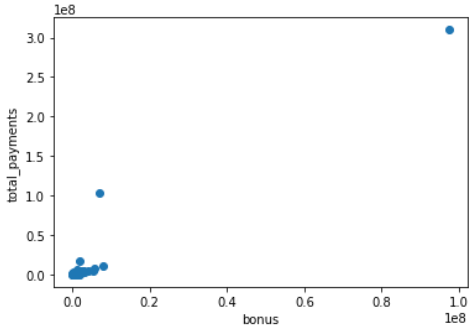
	Vars	NaN	P_Var_NaN
from_messages	86	60	0.410959
other	93	53	0.363014
from_this_person_to_poi	86	60	0.410959
poi	146	0	0.000000
director_fees	17	129	0.883562
deferred_income	49	97	0.664384
long_term_incentive	66	80	0.547945
email_address	111	35	0.239726
from_poi_to_this_person	86	60	0.410959

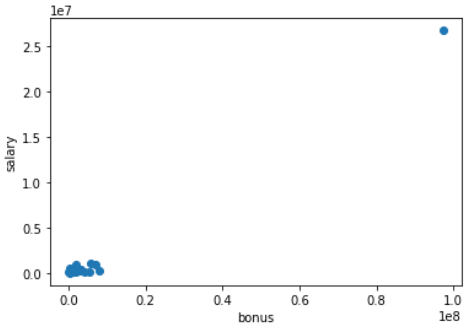
No total são 20 features.

Em resumo, o que contem no conjuntos:

- 146 pessoas, sendo 18 POI e 128 não POI
- 20 características
- 1 Label
- Varias características faltantos mutios dados como foi mostrado na tabela

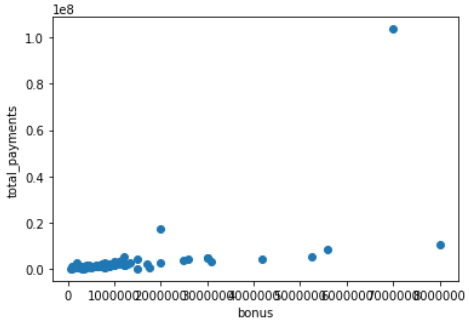
Para os outlies, eu fiz algumas verificações usando gráficos para enchegar esses pontos, segue alguns exemplos:

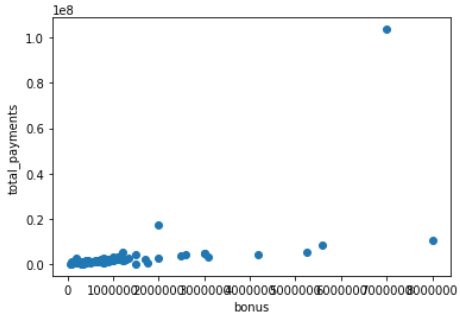




Investigando esse pronto,foi indentificado uma variavel “TOTAL” que estava junto ao cojunto de dados, alem de não ser um nome, estavamuito distante dos outros pontos o mesmo foi removido.

Os mesmo plot após a remoção de “TOTAL”





Investigando esses pontos extremos, os funcinários FREVERT MARK A, LAVORATO JOHN J, WHALLEY LAWRENCE G estão entre esses pontos extremos porém estão classificados como não POI, pode ser que eles sejam de cargo executivo com alto salário, é interessante em retirar essas pessoas.

Em resumo, o que foi retirado do cojunto de dados:

- Variavél “TOTAL”, por não ser uma pessoa e ter um valor muito extremo
- FREVERT MARK A, LAVORATO JOHN J, WHALLEY LAWRENCE G, que são variaveis de valores extremos, porém, não são POI

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”]

Eu usei as seguintes features: ['bonus', 'exercised_stock_options', 'total_stock_value', salary', 'p_bonus', deferred_income',shared_receipt_with_poi','from_poi_to_this_person', 'long_term_incentive', 'p_shared_poi', 'total_payments']

Elas foram selecionadas automaticamente pelo score do SelectKBest dentro do GridSearchCV usando score “recall”, como resultado, o GridSearchCV intendificou que o o paramentro k=13 seria o mais otimizado, como mostra o relatório abaixo:

Pipeline(memory=None,

steps=[('scaler', StandardScaler(copy=True, with_mean=True, with_std=True)), ('selector', SelectKBest(k=13, score_func=<function f_classif at 0x000000000C331828>)), ('svm', SVC(C=1000, cache_size=200, class_weight=None, coef0=0.0,

decision_function_shape='ovr', degree=2, gamma= 0.1, kernel='poly',

max_iter=-1, probability=False, random_state=None, shrinking=True,

tol=0.001, verbose=False))])

	Feature	Score
2	bonus	38.898768
1	salary	25.596746
11	exercised_stock_options	25.349842
14	total_stock_value	24.103581
20	p_bonus	23.829780
3	long_term_incentive	21.175191
4	deferred_income	16.713754
17	from_poi_to_this_person	14.166948
19	shared_receipt_with_poi	14.136210
23	p_shared_poi	10.745080
10	total_payments	10.286364
12	restricted_stock	8.246575
7	other	7.997446

Além seleção de features os dados foram escalonados usando StandardScaler no GridSearchCV, o motivo do escalonamento é devido os dados terem uma grande dispersão.

Foram criados algumas features para o modelo, foram 2 financeiras, 'p_bonus' e 'p_salary', que é razão entre o salario e o total pago, e 'p_bonus', que é a razão do bonus recebido pelo total pago, a ideia era entender quem tinha maior descrepancia entre o que recebe de salário e bonus em relação ao total pago a pessoa, e 2 de emails, 'p_to_poi' e

'p_shared_poi' que era pra criar quais paessos se relacionavam mais com as POI,porem apenas 2 delas ['p_bonus', 'p_shared_poi'] foram selecionadas pelo SelectKBest e entrarem no modelo final, aparetimente a inclusão dessas novas features melhorou a perfomace do algoritimo

| Modelo: SVM | Precision | Recall |

| RF + novas features | 0.40 | 0.39 |

| RF sem novas features | 0.41 | 0.26 |

| Modelo: Random Forest | Precision | Recall |

| RF + novas features | 0.39 | 0.29 |

| RF sem novas features | 0.39 | 0.27 |

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”]

   Eu acabei usando o SVM com kernel 'poly',os algoritmos que foram selecionados para teste foram Random Frorest e SVM, usei a o mapa do Scikit Learn para selecinar os algoritimos.

   | Modelo | Recall | Precision | F1 |

   | Random Forest | 0.39 | 0.29 | 0.33 |

   | SVM | 0.40 | 0.39 | 0.39 |

   Como pode ver, o resultado foram muito próximos no recall mas no precision e F1 teve uma diferença expressiva,dito isso, eu decidir o usar o SVM (Support Vector Machine).

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”]

   Todos os algorimitmos tem os prametros configurados de forma padrão, porém, na maiorias de vezes sempre é possivel mellhorar o desempenho dos algoritimos alterando o os seus parametros manualmente ou automaticamente. Eu usei o GridSearchCV para encontrar automaticamento os paramentros mais otimizados, no caso do SVM que foi o modelo final escolhido, eu configurei os paramentros C, gamma, degree e kernel, segue o resumo dos valores desses paramentros:

- C = [1000]
- Gamma = [0.1]
- Degree = [2]
- Kernel = ['poly']

Esses paramentros foram selecionados dentre outros paramentros automaticamente pelo GridSearchCV o agurmernto 'param_grid' foi inserido da seguinte forma:

```
param_grid = ([{'svm__C': [1,50,100,1000],

        'svm__gamma': [0.5, 0.1],

        'svm__degree':[1,2],

        'svm__kernel': ['rbf','poly'],

        'selector__k':range(1,len(total_features))}])
```

Sendo assim, ele testou todos esses paramentros junto com o classificador e retornou a melhor combinação usando o score "recall" como metrica de referencia.

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"]

   Validação tem como objetivo de garantir a performace de um modelo com cojuntos de dados diferentes do usando no treino, o erro classico de validação é usar os mesmos dados do treinamento para o teste, fazendo isso modelo ficará viciado ou overfitted, com isso, ele irá perfomar muito bem apenas com a base de dados que foi usado para treinar-lo. Existem algums métodos para fazer a validação de um modelo, um deles é fazer o processo de aprendizado supervisionado, que é basicamente dividir o cojunto de dados em duas partes (Treino e Teste), outra maneira é usar a validação cruzada, que é dividir o cojunto em vários subconjuntos menores, o objetivos desses métodos é garantir que o modelo não fique viciado ou overfitted.

   Como o cojunto de dados do projeto é pequeno e desbalanciado (Entre Poi e não POI), eu usei a validação cruzadado GridSearchCV com 5 folds para encontrar os paramentros do classificador, junto com a função fornecida tester.py que usa também validação cruzada com 1000 folds.

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"]

   Precisão: Verdadeiros Positivos / (Verdadeiro Positivos + Falso Positivos). É a

probabildade dos valores que foram classificados como positivos pelo o algorimito estram corretos, ou seja, das pessoas que foram classificadas como POI.

Recall: Verdadeiros Positivos / (Verdadeiros Positivos + Falsos Negativos). É a probabilidade do algoritimo indentificar corretamente todos os intens da umaclasse, ou seja, de todos os POI que existem, quantos foram corretamente classificado.

Há um tradeoff entre a precisão e recal, porem, em casos como este que contém muito mais de uma classe sobre outra classe (maneira mais não-POI do que POI), recall e precisão são melhores medidas do que precisão, mas so porque um modelo tem um precisão muito alta não significa necessariamente que é um ótimo model. Considerando a quantidade de dados faltantes dos features no cojunto de dados a precisão e o recall atingidos é satisfatório.

O desempenho médio do modelo ajustado:

- Precison Score: 0.40
- Recall Score: 0.39
- F1 Score: 0.39