**Resumo do  Projeto Final -  Identify Fraud from Enron Email**

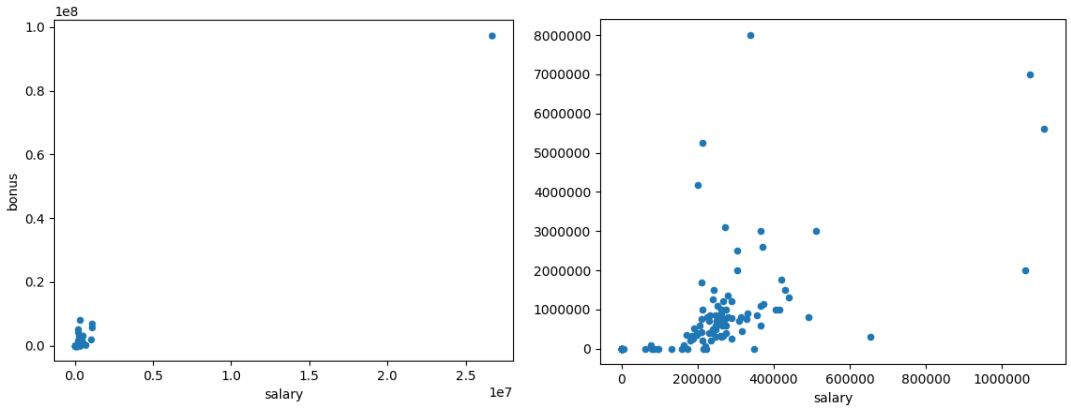
1. Resuma para nós o objetivo deste projeto e como o aprendizado de máquina é útil na tentativa de realizá-lo. Como parte de sua resposta, dê um panorama sobre o conjunto de dados e como ele pode ser usado para responder à pergunta do projeto. Houve algum valor discrepante nos dados quando você os obteve e como você lidou com eles? [rubricas relevantes: “exploração de dados”, “investigação atípica”].

**Resposta:** O objetivo do projeto é, por meio de técnicas de análise de dados e de algoritmos de machine learning,  identificar funcionários da Enron (empresa envolvida num dos maiores escândalos financeiros da história) que podem ter cometido fraude baseando-se no conjunto de dados público entitulado *"Enron financial and email".*  Dessa forma, podem ser utilizados algoritmos de machine learning para, com base nos dados e suas características, tentar encontrar conexões e padrões relacionando os envolvidos.

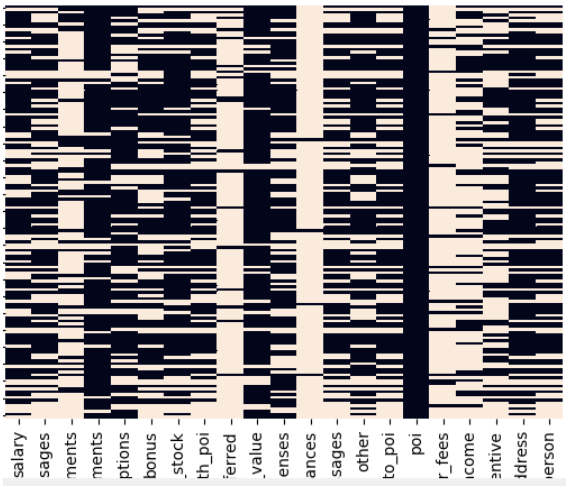
O conjunto de dados possui informações financeiras e informações e relativas à e-mails enviados/recebidos pelos envolvidos.  Essas informações estão organizadas em um conjunto que possui 145 entradas com 20 características.

No conjunto já existem registros de pessoas identificadas como de interesse (POI) num total de 18, restando como não POI a quantidade de 127.

Na busca por valores discrepantes foi encontrado um valor muito acima, quando visualizadas às características salário e bônus, que posteriormente constatou-se que tratava-se da entrada como o nome “TOTAL” que era a linha de somas para cada recurso, a qual foi excluída do conjunto.

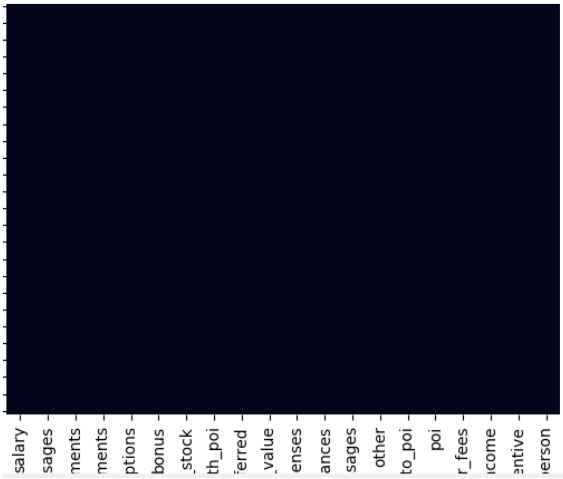


Foram também localizados dados faltantes no conjunto praticamente em todas as entradas à exceção de ‘poi’, conforme mapa abaixo e resultado abaixo.



```
Missing Values:
salary                          51
to_messages                     60
deferral_payments              107
total_payments                  21
exercised_stock_options         44
bonus                           64
restricted_stock                36
shared_receipt_with_poi         60
restricted_stock_deferred      128
total_stock_value               20
expenses                        51
loan_advances                  142
from_messages                   60
other                           53
from_this_person_to_poi         60
poi                              0
director_fees                  129
deferred_income                 97
long_term_incentive             80
from_poi_to_this_person         60
```

Para esses dados faltantes, em relação às características financeiras foi realizada a substituição dos valores faltantes pelo número zero e, em relação às características de e-mail, foi optado por inserir o valor da mediana no lugar dos valores faltantes. Foi removida a característica email_adress por não considerar uma característica importante para o estudo.



2. Quais recursos você usou no seu identificador de POI e que processo de seleção você usou para selecioná-los? Você teve que fazer algum escalonamento? Por que ou por que não? Como parte da tarefa, você deve tentar projetar seu próprio recurso que não vem pronto no conjunto de dados - explique qual recurso você tentou fazer e a lógica por trás dele. (Você não precisa necessariamente usá-lo na análise final, apenas o engenheie e teste.) Na sua etapa de seleção de recursos, se você usou um algoritmo como uma árvore de decisão, forneça também as importâncias de recursos dos recursos que você usa, e se você usou uma função de seleção de recursos automatizada como o SelectKBest, por favor, informe as pontuações dos

recursos e as razões para a sua escolha de valores de parâmetros. [rubricas relevantes: “criar novos recursos”, “selecionar recursos de maneira inteligente”, “dimensionar recursos corretamente”]

**Resposta**: Primeiramente, fazendo uma análise nos dados, foram utilizadas apenas algumas das características financeiras e de e-mail. Foi realizada essa abordagem porque algumas delas visivelmente não resultariam em informação para o projeto (ex: endereço de e-mail).

Não foi utilizado o escalonamento das características em razão de não haver impacto nos algoritmos escolhidos para o projeto.

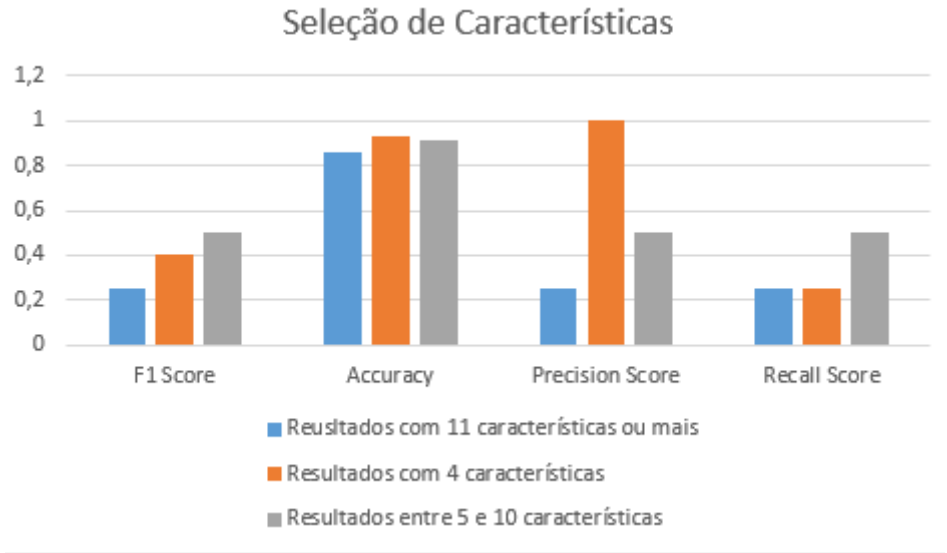
Como nova característica, buscamos trazer um valor que relacionasse os POIs conhecidos e os não POIs . Utilizamos por base uma característica que apresentava uma grande diferença entre as duas classes **('exercised_stock_options')** e criamos um valor da situação atual sobre a média das classes **(df_data['exercised_stock_options'] / float(sum_eso_poi/sum_poi))**, sendo que o resultado foram duas novas características contrapostas **('fraction_exercised_stock_options_poi' e 'fraction_exercised_stock_options_npoi').**

Pode-se verificar durante a execução do script que as novas características tiveram um bom desempenho, pois foram selecionadas pelo recurso utilizado entre as cinco melhores caracteríscas.

```
('Features list:', ['poi', 'bonus', 'total_stock_value', 'exercised_stock_options', 'fraction_exercised_stock_options_poi', 'fraction_exercised_stock_options_npoi'])
```

Contudo, quando retiradas do conjunto o algoritmo obteve um resultado semelhante.

Para seleção das características, foi utilizado o recurso *SelectKBest*, em várias iterações, e foi o obtido o resultado conforme gráfico abaixo**:**




Os resultados quando utilizadas entre 5 e 10 características foram os que obtiveram os resultados mais consistentes com os objetivos do projeto.

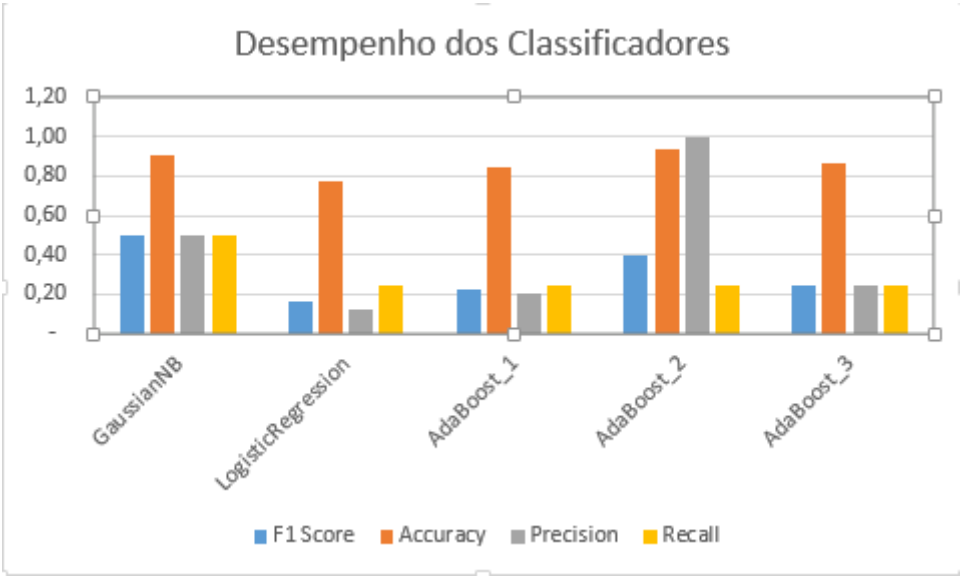
Diante disso, foi optado pela utilização de 5 características além de 'poi', restando apenas: **['poi', 'salary', 'bonus', 'exercised_stock_options', 'fraction_exercised_stock_options_poi', 'fraction_exercised_stock_options_npoi']**

3. Qual algoritmo você acabou usando? Que outro (s) você tentou? Como o desempenho do modelo diferiu entre os algoritmos? [rubrica de rubrica relevante: "escolha um algoritmo"]

**Resposta:** Foram utilizados três algoritmos: GaussianNB, LogisticRegression e AdaBoostClassifier, sendo que nesse último foram testadas três configurações diferentes e os resultados obtidos estão listados na tabela abaixo:

| scores / classificadores | GaussianNB | LogisticRegression | AdaBoost_1 | AdaBoost_2 | AdaBoost_3 |
|---|---|---|---|---|---|
| F1 Score | 0.5 | 0.16 | 0.2222 | 0.4 | 0.25 |
| Accuracy | 0.9090 | 0.7727 | 0.8409 | 0.9318 | 0.8636 |
| Precision | 0.5 | 0.125 | 0.2 | 1 | 0.25 |
| Recall | 0.5 | 0.25 | 0.25 | 0.25 | 0.25 |

Conforme resultados registrados na tabela acima, pode-se verificar que o melhor desempenho para o projeto, levando-se em consideração os índices mais importantes, foi o do classificador GaussianNB.




O classificador AdaBoost_2 também teve um bom desempenho, contudo ele não foi muito bem em um indicador importante para o projeto (Recall). Dessa forma, classificador que obteve um resultado mais equilibrado foi o GaussianNB.

4. O que significa ajustar os parâmetros de um algoritmo e o que pode acontecer se você não fizer isso bem? Como você ajustou os parâmetros do seu algoritmo particular? Quais parâmetros você ajustou? (Alguns algoritmos não possuem parâmetros que você precisa ajustar - se este for o caso daquele que

você escolheu, identifique e explique brevemente como você teria feito isso para o modelo que não foi a sua escolha final ou um modelo diferente que faz utilizar o ajuste de parâmetros, por exemplo, um classificador da árvore de decisão). [rubricas relevantes: “discutir o ajuste dos parâmetros”, “ajustar o algoritmo”]

**Resposta:** Os parâmetros são alguns recursos que estão disponíveis para determinados algoritmos e servem para adequar o algoritmo ao problema e às características específicas de cada conjunto de dados.

A utilização dos parâmetros afeta no resultado e pode tanto melhorar como piorar o desempenho do algoritmo. No projeto, pode-se verificar que no algoritmo AdaBoost foram utilizadas três configurações diferentes, sendo que na última (que teve o um desempenho melhor que a primeira, contudo, inferior ao GaussianNB) foi utilizado, para o tuning do modelo, o recurso GridSearchCV, com tipos de parâmetros diferentes. No algoritmo **AdaBoost_3** foram testados os seguintes parâmetros **{'n_estimators': [2,4,6], 'learning_rate' :[0.1, 0.5, 1., 10], 'base_estimator__min_samples_split' : np.arange(2, 8, 2),'base_estimator__max_depth' : np.arange(1, 4, 1)}.** Esses parâmetros eram direcionados ao algoritmo base_estimator DecisionTreeClassifier e basicamente atuavam na profundidade e divisão em relação ao modo de classificação.

5. O que é validação e qual é um erro clássico que você pode cometer se errar? Como você validou sua análise? [rubricas relevantes: “discutir validação”, “estratégia de validação”]

**Resposta:** A validação é um procedimento que busca avaliar se o algoritmo realmente funciona. Geralmente, divide-se o conjunto de dados em um conjunto de treino e outro de teste e, dessa forma, verifica-se o desempenho do algoritmo.

O que se quer evitar é que o algoritmo aprenda em excesso com os dados (decorar os dados) o que pode causar o sobreajuste, sendo que nesse caso o algoritmo terá um ótimo resultado no treinamento, mas na fase de teste o desempenho cai porque o algoritmo perde sua capacidade de generalizar. O contrário também pode acontecer, ou seja, um sub-ajuste, sendo que nesse caso o algoritmo é muito simplista e não consegue ter um bom desempenho no teste.

No projeto foi utilizada a ferramenta *sklearn.model_selection.train_test_split* para dividir o conjunto em partes de treino e teste, utilizando o tamanho de 0,3 para teste que resultou em 100 entradas para treino e 44 entradas para teste. Existem formas mais aperfeiçoadas de validação como a validação cruzada que significa ter mais um conjunto de dados além dos conjuntos de treinamento e de teste. Dessa forma o conjunto de treinamento é usado para treinar os parâmetros, o conjunto de validação cruzada é utilizado para tomar as decisões sobre o modelo e o conjunto de teste é utilizado para o teste final do modelo.

6. Dê pelo menos duas métricas de avaliação e seu desempenho médio para cada uma delas. Explique uma interpretação de suas métricas que diz algo que seja compreensível ao homem sobre o desempenho do seu algoritmo. [rubrica de rubrica relevante: “uso de métricas de avaliação”]

**Resposta:** Foram utilizadas quatro métricas para avaliar os resultados dos algoritmos**: Accuracy ( acurácia), Precision Score (precisão), Recall Score (revocação) e F1 Score.**

O algoritmo obteve acurácia de 0.9090. Esse indicador informa a frequência de quanto o algoritmo está correto, ou seja, a soma dos verdadeiros positivos e verdadeiros negativos sobre o total.

A precisão é um indicador que busca informar o quanto de verdadeiros positivos estavam corretos (TP)/(TP +FP). O nosso algoritmo obteve 0.5. Devido sua característica essa é uma métrica interessante para o projeto.

Recall é uma métrica que indica a frequência em que algoritmo encontra os exemplos de uma classe. Nosso algoritmo obteve 0.5. Também é uma métrica importante para avaliação do resultado projeto.

O F1 Score é uma métrica que combina precisão e recall que mostra de forma única a qualidade geral do modelo. Nosso algoritmo obteve o resultado de 0.5, sendo que o máximo seria um (1/1).

Consideração: Devido às características do projeto acredito que as duas métricas principais para se obter um bom resultado seriam precisão e recall. O recall identificaria pessoas possivelmente envolvidas para uma primeira análise em razão da divisão das classes, mas necessitaria de um trabalho posterior de investigação. A precisão informa que as pessoas indicadas como POI teriam grande probabilidade de serem POIs.