Enron Submission Free-Response Questions

1 - O objetivo desse projeto foi analisar os clássicos dados da empresa Enron, que se envolveu em uma confusão por ser acusada de fraude fiscal nos Estados Unidos. Com esses dados foi necessário achar a melhor técnica de Machine Learning para que se pudesse obter as pessoas de interesse que estavam envolvidas no escândalo da empresa. A análise poderia ter sido feita de varias formas, mas como os dados são muitos, pois muitas pessoas trabalhavam na empresa e nem todas estavam envolvidas nas fraudes, e como existem muitas *features* em que podemos tirar conclusões, como todos os emails enviados na empresa e *features* sobre os salários e bônus pagos pela empresa, foi utilizado o Machine Learning que consegue analisar todos os dados com todas as variáveis possíveis de uma vez e nos dar um resultado muito bom do que de melhor se pode tirar dos dados.

Como a base de dados é muito grande, muitas vezes os dados não vem totalmente corretos. Portanto foi necessário fazer uma limpeza antes de começar a análise. Lendo o documento da empresa “Enron Insider Pay”, foi observado dois *outiliers* que precisam ser removidos, sendo eles o total de cada valor pago aos funcionários e os valores pagos a uma empresa de nome “The travel Agency in the park” de uma empresa irmã da Enron que gerenciava viagens. Também foraram alterados valores nulos para o valor 0 para base de cálculos.

2 - Para uma primeira análise foram utilizadas todas as *features* disponíveis, tanto as de pagamentos como as de email, também foram adicionadas duas mais *features* ao conjunto, foram elas “fraction_from_poi" e “fraction_to_poi”, elas foram obtidas pelo resultado da divisão dos emails que foram enviados para uma pessoa de interesse pelo total de emails e dos emails que foram recebidos por uma pessoa de interesse pelo total de emails. Para as divisões que obtinham um valor zero, foi considerada o resultado da divisão como zero também. Depois de testado os classificados com todas as *features* foi utilizado o SelectPercentile para reduzir as *features* para 20%, sendo elas as que obtiveram melhor resultado nos classificadores.

3 - Para a análise foram testados três algoritimos, sendo eles o Naive Bayes, Decision Trees e o KNeighbors. O modelo que teve um melhor desempenho foi o KNeighbors, tendo um "Accuracy score" de 0.89, o algoritmo Naive Bayes teve um desempenho bem baixo utilizando todas as *features* e Decision Trees teve um valor máximo de 100%, ou seja, alguma coisa não estava certa. Depois de ajeitada as *features* foi possível obter valores melhores dos algoritmos, agora o melhor desempenho foi o Naive Bayes com “Accuracy Score” de 0.88, ja o Decision Trees “melhor" seu desempenho com 0.79 e o KNeighbors com 0.86. Por final o algoritmo escolhido com o KNeighbors por alem do “Accuracy Score” ter mantido um bom desempenho, existiram outros parâmetros que desempenharam melhor.

4 - Quando é preciso tunar os parâmetros quer dizer que é preciso passar características para o algoritmo para que ele possa desempenhar melhor. Para isso existe uma serie de opções que podem ser passadas para o classificador de acordo com a sua documentação para voce analisar e testar qual é a melhor forma de desempenhar o classificador. Se não for feito isso você obterá um modelo que talvez não seja o que melhor descreva os dados reais, ou seja os parâmetros de avaliação vão ser menores. Para melhorar o algoritmo foi utilizado técnica como o SelectPercentile com um parâmetro de seleção de 30% para as melhores *features*.

5 - A validação é feita através do train_test_split, onde as *features* e os *labels* são separados em dados de treino e de teste e é feito a avaliação do classificador. Ele é utilizado para que o classificador não tenha grande variância e também não fiquei muito enviesado. No train_test_split foram utilizado paramelhor como dados de teste em 30% e um random_state de 42.

6 - As métricas para avaliar os classificadores escolhidas foram “Acuracy Score”, “Recall Score” e “F1 Score”. Quanto maior a acurácia do classificador quer dizer que ele esta o mais próximo de um modelo ideal, o Recall Score seria o numero de positivos no modelo, ou seja de pessoas que seriam de interesse e o F1 Score seria a métrica dos números falsos positivos e e falsos negativos.