Enron Submission Free-Response Questions

A critical part of machine learning is making sense of your analysis process and communicating it to others. The questions below will help us understand your decision-making process and allow us to give feedback on your project. Please answer each question; your answers should be about 1-2 paragraphs per question. If you find yourself writing much more than that, take a step back and see if you can simplify your response!
When your evaluator looks at your responses, he or she will use a specific list of rubric items to assess your answers. Here is the link to that rubric: Link to the rubric Each question has one or more specific rubric items associated with it, so before you submit an answer, take a look at that part of the rubric. If your response does not meet expectations for all rubric points, you will be asked to revise and resubmit your project. Make sure that your responses are detailed enough that the evaluator will be able to understand the steps you took and your thought processes as you went through the data analysis.
Once you've submitted your responses, your coach will take a look and may ask a few more focused follow-up questions on one or more of your answers.
We can't wait to see what you've put together for this project!

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 deste Projeto era identificar funcionários da Enron que poderiam ter cometido crime de fraude com bases nos dados financeiros e e-mails da Enron. Os dados financeiros possuíam informações sobre salários, bônus e ações de diversos funcionários. A base com os e-mails possuíam uma lista de todos os e-mail enviados entre os funcionários no período da fraude. Também tinha uma lista com quais funcionários foram condenados ou investigados. Com essas informações foi possível analisar uma correlação entre salários x bônus, salários x valor das ações e e-mail trocados de suspeitos e não suspeitos para identificar um padrão de comportamento entre funcionários suspeitos em não suspeitos.

   A base de dados possui 146 pessoas, cada uma com 21 características. As características foram financeiras (ex. salário e bônus) e email (ex. número de mensagens enviadas e recebidas). Das 146 pessoas da base de dados, 18 são POI´s. As características mais importantes são "Total_payments", 'deferred_income' e 'long_term_incentive', 'deferral_payments', 'bonus' e 'expenses'.

   Existem alguns problemas nessa base como outlier e valores faltantes. Os outliers foram identificados através de scatter plots e validados através do pdf que deu origem aos dados financeiros e após isso foram removidos da base de dados. Por exemplo, ao se plotar um gráfico de salario vs bônus foi possível verificar um funcionário com valores muito acima dos demais. Ao se investigar o pdf para verificar qual funcionário era esse, foi identificado que era o valor total não um funcionário propriamente dito. Também foi identificado uma empresa "The Travel Agency Park' que não possuía valores de ações e salários. Ela também foi removida da base. Eu removi o Eugene Lockhart porque notei que todas as características deles eram nulas ou 0.

Existiam muitos valores faltantes. Características como “loan_advances”, “restricted_stock_deferred” possuem muitos valores faltantes e serão invalidadas. “Características como Other”, e “diretor_fees” tem pontos para poucas pessoas já que se referem a pagamentos muito particulares e por isso também irei remove-las do data_set. Os valores faltantes de outras características foram substituídos com a mediana da cada uma das características. A mediana foi utilizada pois ela exclui os valores muito baixos e muito altos que podem descaracterizar a média principalmente considerando que existem algumas pessoas com salários muito acima.

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”]

Para selecionar as características que seriam utilizadas nos algoritmos gerei vários gráficos cruzando várias características com o Bônus. Com isso foi possível identificar quais características mostravam algum tipo de correlação com o Bônus e quais não mostravam. Também foi possível identificar características que tinham poucos pontos de dados (muitos valores faltantes) e com isso seria difícil encontrar algum padrão através delas. Após isso verifiquei quais características poderiam ser geradas e que não estavam no data set. Criei um índice com a relação de e-mail enviados das pessoas para suspeitos ou recebidos de suspeitos sobre o número total de e-mails enviados. Se o número fosse alto poderia indicar algum indicio de que essa pessoa também seria um suspeito.

Foi necessário aplicar um escalonamento de característica utilizando “MinMaxScale”. Existiam valores financeiros com uma variância muito diferente entre elas e algumas características como “Número de e-mail enviados” tinham valores muito menores do que os valores financeiros. Isso causou muita demora nos algoritmos e os resultados foram ruins. Como ultima etapa na seleção de característica eu utilizei o algoritmo “Kbest”. Eu apliquei o Kbest no algoritmo “GridSearchCV” para identificar o melhor valor de K. O algoritmo retornou que o melhor valor de K é 12. As características selecionadas e seus scores são as abaixo:

{'salary': 0.0099823995896920984, 'deferral_payments': 8.9591366476908512, 'total_payments': 30.728774633399691, 'bonus': 8.7922038527052457, 'total_stock_value': 4.1807214846470675, 'expenses': 9.6800414303809781, 'from_poi_to_this_person': 0.11120823866694529, 'exercised_stock_options': 7.5551197773202912, 'from_this_person_to_poi': 1.7429798402418903, 'poi': 15.858730905995106, 'deferred_income': 10.633852048382536, 'shared_receipt_with_poi': 0.11120823866694529, 'restricted_stock': 4.9586666839661397, 'long_term_incentive': 8.0583063122804752}

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 utilizei o algoritmo Naïve Bayes (GaussianNB). Eu também tentei o Random Forest, SVM e Decision Tree mas o Naive Bayes foi o que melhor performou. O Naive bayes não foi o algoritmo com melhor acurácia, Random Forest e SVM foram melhor nesse quesito. No entanto, o Naive Bayes foi melhor em precision e recall. Alguns algoritmos foram muito bem em Precision mas muito mal no recall como Random Forest e SVM. Decision Tree também chegou em bons resultados, bem próximos aos do Naive Bayes. Segue abaixo a performance de cada algoritmo:

Naive Bayes:

Accuracy: 0.83013 Precision: 0.35729 Recall: 0.34300 F1: 0.35000 F2: 0.34577

Random Forest:

Accuracy: 0.84720 Precision: 0.32864 Recall: 0.14000 F1: 0.19635 F2: 0.15816

SVM

Accuracy: 0.85407 Precision: 0.33508 Recall: 0.09600 F1: 0.14924 F2: 0.11198

DT

Accuracy: 0.81840 Precision: 0.31224 Recall: 0.30100 F1: 0.30652 F2: 0.30318

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”]

Afinar um algoritmo significa ajustar variáveis/parâmetros que podem alterar a performance do algoritmo de forma a fazê-lo performar melhor. Se ao afinar o algoritmo você não tiver cuidado ao escolher os parâmetros você pode levar o algoritmo ao erro como a sobreposição. Isso significa que o algoritmo irá se ajustar demais e aprender desvios específicos que não deveriam ser considerados. Um modelo com sobreposição possui alta precisão mas não reflete a realidade.

O algoritmo que escolhi não precisa de parâmetros mas eu utilizei alguns outros como o SVM onde foi necessário realizar ajustar alguns parâmetros. No SVM utilizei dois parâmetros: “Kernel” e o “C”.

O “C” é um parâmetros para ajustar a sobreposição. Baixos valores de C o modelo irá ser pouco sensível ao fazer o ajuste, ou seja, irá se basear pelo padrão. Para altos valores de C o algoritmo irá ser muito sensível e se ajustar aos menores desvios.

Kernel é uma forma do algoritmo de fazer cálculos mais rápidos quando se tem dados com muitas dimensões. Ajustando esse parâmetro você pode dizer para o algoritmo não utilizar uma forma linear de separar seus dados mas tentar de outras formas não lineares de transformar os dados.

Para encontrar o melhor parâmetro eu utilizei o algoritmo “GridSearchCV” onde passei uma lista de possíveis valores para que ele testasse todos e retornasse o com melhor resultado. Para esse caso os melhores parâmetros encontrados foram C = 100 e Kernel = Linear

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”]

Validação é um processo quando validamos nosso modelo de treinamento, testando-o em uma base de dados ainda não utilizada. A forma padrão de fazer isso é dividimos nossa base de dados em base de treino e de teste (70%, 30% respectivamente). Dessa forma executamos o algoritmo com a base de treino e vemos o quanto ele acertou utilizando a base de teste. No entanto se essa divisão da base entre treino e teste não for bem feita, isto é, não forem amostras heterogêneas ou forem muito pequena, o resultado final poderá ser de sobreajuste.

Como o conjunto de dados era muito pequeno para termos uma base de validação totalmente independente dos conjuntos de treinamento e teste utilizei o método de validação cruzada StratifiedShuffleSplit com 1000 folds e random_state de 42 conforme script tester.py para validar meu modelo. O StratifiedShuffleSplit divide várias vezes o dataset principal em bases de teste e treinamento independentes. O parâmetro random state controla a aleatoriedade na geração dessas bases. O parâmetro fold controla a quantidade de amostras aleatórias serão geradas.

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”]

Naive Bayes:

Accuracy: 0.83440. Isso significa que meu modelo estava 83,44% certo ao identificar se uma pessoa era ou não um POI. Esta métrica não era necessariamente importante neste modelo pois se o algoritmo tivesse previsto que todas as pessoas eram não-POI, a acurácia seria de 87,5%.

Precision: 0.38114. Isso significa que em 38,11% das vezes que meu modelos classificou uma pessoa como POI ela era efetivamente um POI. Então o calculo POI_verdadeiros/(POI_falsos + POI_verdadeiros) foi de 38,11%. Ele é importante pois quanto mais alto a precisão menos a quantidade de falsas acusações comparadas com as acusações verdadeiras.

Recall: 0.38800 . Isso significa que 38,80% dos suspeitos foram identificados corretamente pelo algoritimo. A equação para essa métrica é: POI_verdadeiros/(non_POI_falsos + POI_verdadeiso. È uma métrica importante pois indica que acertamos o máximo possível ao identificar os POIs.