Projeto Identificando uma fraude através dos emails e dados financeiros da Enron

Roberto da Conceição Santos
Udacity - Fundamentos de Data Science II

A empresa Enron era uma empresa americana de energia, commodities e serviços com sede em Houston, Texas, que foi uma das maiores empresas dos Estados Unidos. Em 2002, a empresa entrou em colapso devido à grande quantidade de fraudes corporativas e contábeis. Seu colapso afetou milhares de funcionários e influenciou todo o sistema econômico ocidental.

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 é usar dados financeiros e de e-mail dos executivos da empresa Enron, que foram liberados pelo governo dos EUA após a investigação efetuada, para chegar a um modelo preditivo que possa identificar pessoas possivelmente envolvidas na fraude identificada neste projeto como POIs. O machine learning pode ser uma ferramenta útil nesse tipo de situação, pois com a utilização de algoritmos pode identificar padrões nos dados e realizar treinos para reconhecer esses padrões.
   O dataset usado neste projeto possui 146 registros, 21 características disponíveis, sendo 14 financeiras (pagamentos e investimentos), 6 de e-mail e 1 rótulo (se é um POI); dos 146 registros 18 são POIs e 128 são non-POIs; durante a análise foram identificados características com muitos valores 'NaN' conforme mostra a tabela abaixo:

Característica	Qtde.NaN
nome	0
bonus	64
deferral_payments	107
deferred_income	97
director_fees	129
email_address	35
exercised_stock_options	44
expenses	51
from_messages	60
from_poi_to_this_person	60
from_this_person_to_poi	60
loan_advances	142
long_term_incentive	80
other	53
poi	0
restricted_stock	36
restricted_stock_deferred	128
salary	51
shared_receipt_with_poi	60
to_messages	60
total_payments	21
total_stock_value	20

   Ao analisar os dados, me deparei com um outlier (dado discrepante) para um funcionário com nome TOTAL, que mais parecia ser o totalizador dos salários, causado por algum erro na criação do dataset. Este outlier "Total" foi removido do dicionário de dados usando a linha "data_dict.pop ('TOTAL', 0)".

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”]

   Inicialmente comecei a usar as seguintes features: ‘salary’, 'bonus', 'shared_receipt_with_poi', 'from_this_person_to_poi', 'from_poi_to_this_person', 'expenses'. Estas features foram escolhidas sob os

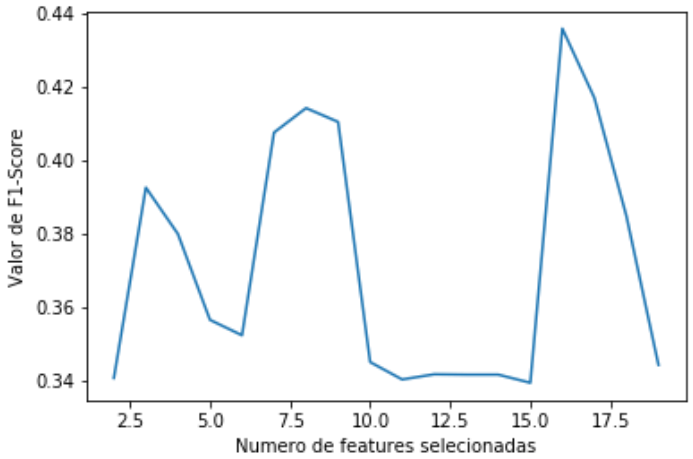
pressupostos de que a maioria dos POIs teria um conexão (relacionamento) com outros POIs e também algum padrão apareceria em seus salário e dados de bônus. A nova feature que eu olhei foi a soma do número de e-mails para cada pessoa. Ou seja, "from_this_person_to_poi" + "From_poi_to_this_person".
A principal razão para esta escolha foi a possibilidade de um ligação com um POI com este número total de emails trocados.
O dimensionamento de features foi implementado na minha análise principalmente porque o intervalo de valores para 'salary', 'bonus' and 'expenses' eram altos quando comparados aos de outras features.
Em última análise, apenas as três features mais importantes foram usadas com a utilização do algoritmo Decision Tree (DT). Seguem os valores de importância que obtive para as features que usei.

Feature = 'expenses', Importância = 0.46238
Feature = 'shared_receipt_with_poi', Importância = 0.28012
Feature = 'from_this_person_to_poi', Importância = 0.21334

Métricas de performace com e sem as novas features:

Modelo	Acurácia
NB com novas features	0.640
NB sem novas features	0.792
DT com novas features	0.840
DT sem novas features	0.833
SVM com novas features	0.720
SVM sem novas features	0.750

Foi utilizado o SelectKBest para indicar a melhor quantidade de features que serão utilizadas no modelo final. Para a escolha do número final, foi observado a variação do Score F1.



Foi utilizado MinMaxScaler para definir o menor valor para 0 e o maior como 1 para todas as features. Foi necessário escalar as features, devido a quantidade de outliers que poderiam prejudicar o modelo sendo escolhidos como preditores principais. Outro motivo importante para o escalonamento, é que as features com valores ausentes foram preenchidos com 0 e o MinMaxScaler define como 0 os menores valores.

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"]

   Neste projeto foram testados três algoritmos: Decision Tree (DT), Naive Bayes (NB) e Support Vector Machines (SVM). Na análise final foi utilizado o Decision Tree porque este apresentou os melhores valores de acurácia, precisão e recall. Seguem abaixo os desempenhos de cada algoritmo.
   DT : Acurácia = 0.833, Precisão = 1.00, Recall = 0.2
   NB : Acurácia = 0.792, Precisão = 0.00, Recall = 0.00
   SVM : Acurácia = 0.750, Precisão = 0.00, Recall = 0.00

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”]

Tunar um algoritmo se refere ao processo de buscar o melhor ajuste para obter os melhores resultados. Alguns parâmetros em alguns algoritmos precisam de valores como entradas do usuário, pois esses valores não podem ser obtidos ou inferidos a partir do próprio conjunto de dados. Estes valores de parâmetros precisam ser cuidadosamente escolhidos para não ocorrer o risco de comprometer o desempenho do algoritmo. o tuning foi realizado com o GridSearchCV do pacote sklearn que a partir de um conjunto de parâmetros escolhidos, realiza um cruzamento com todas as combinações possíveis para encontrar o melhor ajuste. Nesta análise foram usados os parâmetros:

SVM : ‘C’, ‘gamma’

DT : ‘min_samples_split’, ‘min_samples_leaf’

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 é o processo de verificar o qual bem o seu modelo irá generalizar para novos dados. Um erro clássico é treinar o modelo com todos os dados, fazendo com que o modelo memorize a classificação e não aprenda a generalizar. Nessa análise, o algoritmo foi treinado usando apenas 80% do conjunto de dados e foi validado usando os 20% restantes do conjunto de dados como o conjunto de testes. O 'train_test_split' do pacote 'cross_validation' em sklearn foi usado para dividir os dados em conjuntos de treinamento e testes.

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”]

   Métricas obtidas em cada algoritmo testado:
   DT : Acurácia = 0.833, Precisão = 1.00, Recall = 0.2
   NB : Acurácia = 0.792, Precisão = 0.00, Recall = 0.00
   SVM : Acurácia = 0.750, Precisão = 0.00, Recall = 0.00

   O modelo final selecionado possui acurácia = 0.799, Precisão = 0.3805, Recall = 0.3255, neste contexto, tem se uma precisão de 0,38; ou seja, em 38% dos casos o modelo classifica os funcionários como POI. O recall significa que para todos os casos de POI reais, o modelo classifica 33% como POI. O F1 é a média harmônica de precisão e recall onde F1 = 2 * (precisão * recall) / (precisão + recall).