Projeto de Machine Learning

Identificar Fraude nos E-Mails da Enron

1. Resuma para nós o objetivo deste projeto e como o machine learning 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 de outliers"]

O objetivo deste projeto foi utilizar, testar e comparar diferentes algoritmos de machine learning para identificar padrões na base de dados da Enron. Machine learning é um ramo da inteligência artificial baseado na idéia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana.

A base de dados original da Enron contém 146 linhas (registros) e 20 colunas (características ou recursos ou features). Nessas colunas, além da indicação POI/não-POI, incluem dados financeiros (salário, bônus, despesas, valor das ações, etc.) e também dados de e-mails enviados/recebidos (número de mensagem enviadas para um POI, número de mensagens recebidas de um POI, etc.). Desses 146 registros, 18 estavam identificados como POI, ou seja, pessoas suspeitas de terem participado na fraude da empresa.

Antes de utilizar esses dados nos algoritmos de machine learning, esses dados passaram por um etapa de exploração para identificação dos dados ausentes e discrepantes, também denominados outliers.

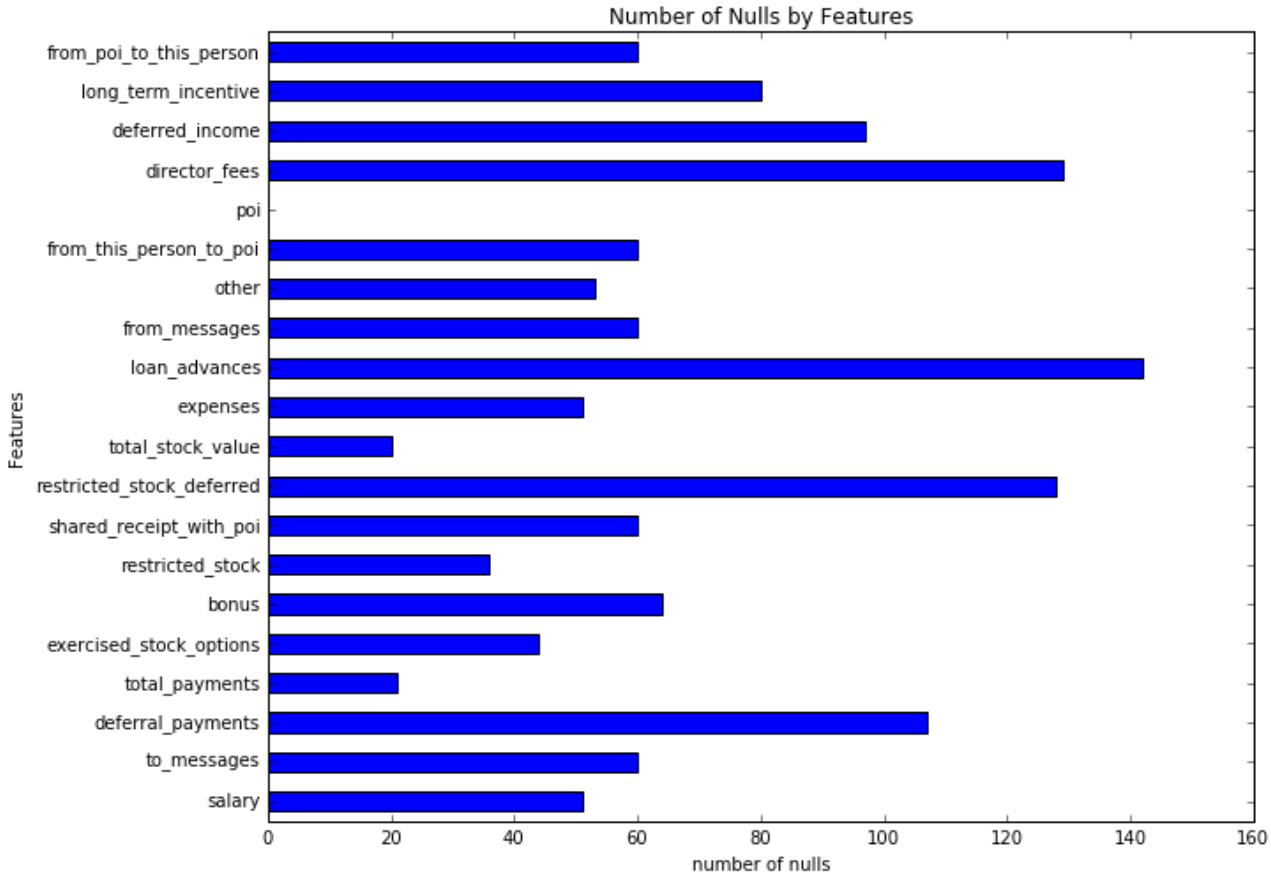

Figura 1 – Quantidade de valores nulos por recurso

Tabela 1 – Quantidade de valores nulos por recurso

Features	Qtdade de Nulos
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

Observa-se que a coluna “loan_advances” possui apenas 3 linhas não nulas (“FREVERT MARK A”, “LAY KENNETH L” e “PICKERING MARK R”). Dessas linhas, apenas “LAY KENNETH L” está definido como POI. Então, devido ao grande quantidade de valores nulos e pouca relevância, esta coluna foi excluída.

Como analisado em sala de aula (ver Aula 12.4), foram identificados 3 outliers e seus dados foram excluídos da base de dados original. As Figuras 2 e 3 exibem antes e depois da exclusão da linha “TOTAL” (tratava-se uma linha de totalização dos registros da base de dados).

Outras linhas excluídas, foram: “LOCKHART EUGENE E”, devido à ausência de dados significativos (presença apenas valores nulos) e “THE TRAVEL AGENCY IN THE PARK”, por não se referir à pessoa física.

A caracteristica “email_address” tambem foi excluída, visto que o conteudo e texto e a analise deste projeto esta baseada nos dados financeiros e na quantidade de e-mails trocados entre as pessoas.

Ao final desta etapa, os valores nulos foram substituídos por zero.

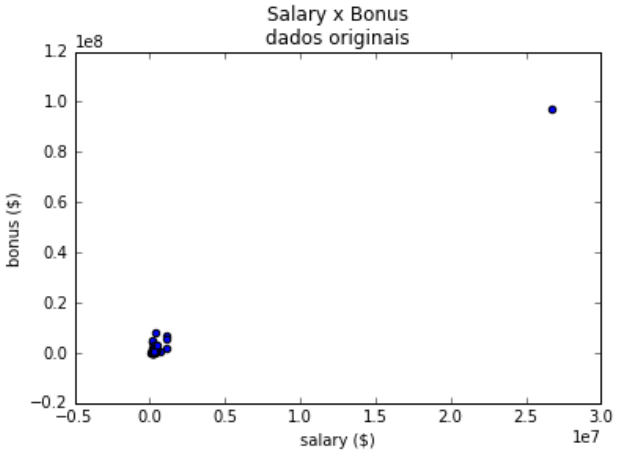

Figura 2 – Visualizacao dos dados antes da exclusao da linha "TOTAL"

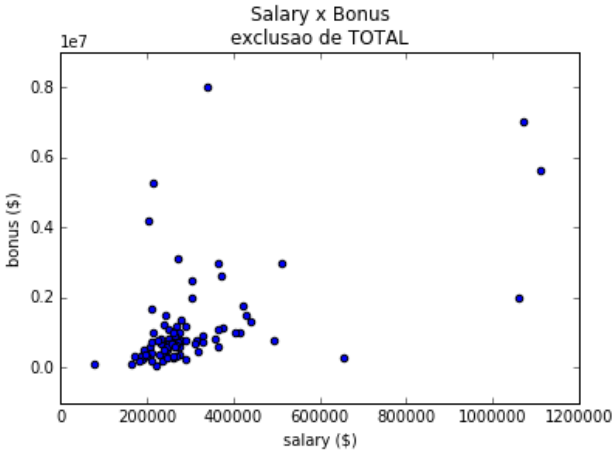

Figura 3 – Visualizacao dos dados depois da exclusao da linha "TOTAL"

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 crie 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"]

Conforme Aula 12.4, "fraction_from_poi" e "fraction_to_poi" foram dois recursos (features) criados em sala de aula e que revelaram possuir padrões possíveis de serem explorados:

- fraction_from_poi: número percetual de mensagens recebidas de POI (from_poi_to_this_person / to_messages)
- fraction_to_poi: número percetual de mensagens enviadas de POI (from_this_person_to_poi / from_messages)

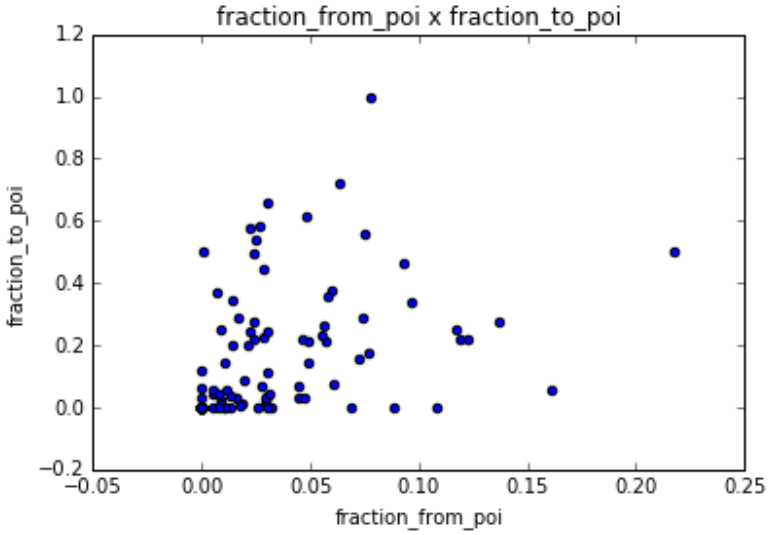

Figura 4 – Gráfico com novos recursos: fraction_from_poi x fraction_to_poi

Na Tabela 5 (ver próximo item), para efeito de comparação, são apresentados os resultados dos modelos com os novos recursos e sem esses recursos.

Além disso, nesta etapa foram utilizadas as seguintes funções:

- MinMaxScaler: dimensionamento de cada recurso para um determinado intervalo (ajuste de escala). Os principais motivos em se aplicar esse dimensionamento é que o intervalo dos dados originais pode ser muito grande e isso pode influenciar o gradiente no processo de convergência.
- SelectKBest: identificação e seleção dos melhores recursos capazes de prever a classe ou como fator de regressão

Tabela 2 – Classificacao dos melhores recursos: score e p-value

Order	Feature	Score	P_Value
1	exercised_stock_options	24,81508	0,00000
2	total_stock_value	24,18290	0,00000
3	bonus	20,79225	0,00001
4	salary	18,28968	0,00003
5	fraction_to_poi	16,40971	0,00008
6	deferred_income	11,45848	0,00092
7	long_term_incentive	9,92219	0,00199
8	restricted_stock	9,21281	0,00286
9	total_payments	8,77278	0,00359
10	shared_receipt_with_poi	8,58942	0,00395
11	expenses	6,09417	0,01476
12	from_poi_to_this_person	5,24345	0,02351
13	other	4,18748	0,04258
14	fraction_from_poi	3,12809	0,07912
15	from_this_person_to_poi	2,38261	0,12493
16	director_fees	2,12633	0,14701
17	to_messages	1,64634	0,20156
18	deferral_payments	0,22461	0,63628
19	from_messages	0,16970	0,68100

Os recursos com p-values inferiores a α = 0,05 significam que esses recursos influenciam o modelo (falharam em rejeitar a hipótese nula).

Neste projeto, foram testados os algoritmos: Decision Tree, SVC e AdaBoost.

Após a etapa de ajuste, foram realizadas simulações com a função RFECV a fim de exibir o número de recursos mais relevantes para cada modelo.

Tabela 3 – Simulacao com variacao dos recursos utilizados

Num. Features	F1 Score		
	Decision Tree	SVC	AdaBoost
1	0,3826	0,2001	0,1481
2	0,3826	0,2461	0,1333
3	0,3826	0,2052	0,2952
4	0,3826	0,1790	0,3185
5	0,3826	0,3264	0,3317
6	0,3181	0,3157	0,4333
7	0,3212	0,2601	0,3333
8	0,4434	0,3240	0,3333
9	0,2885	0,2778	0,3712
10	0,2610	0,1626	0,3333
11	0,2314	0,1606	0,4038
12	0,3852	0,1520	0,4038
13	0,5438	0,1520	0,4038

A Figura 5 apresenta o desempenho de cada algoritmo a partir da variação do número de recursos no modelo.

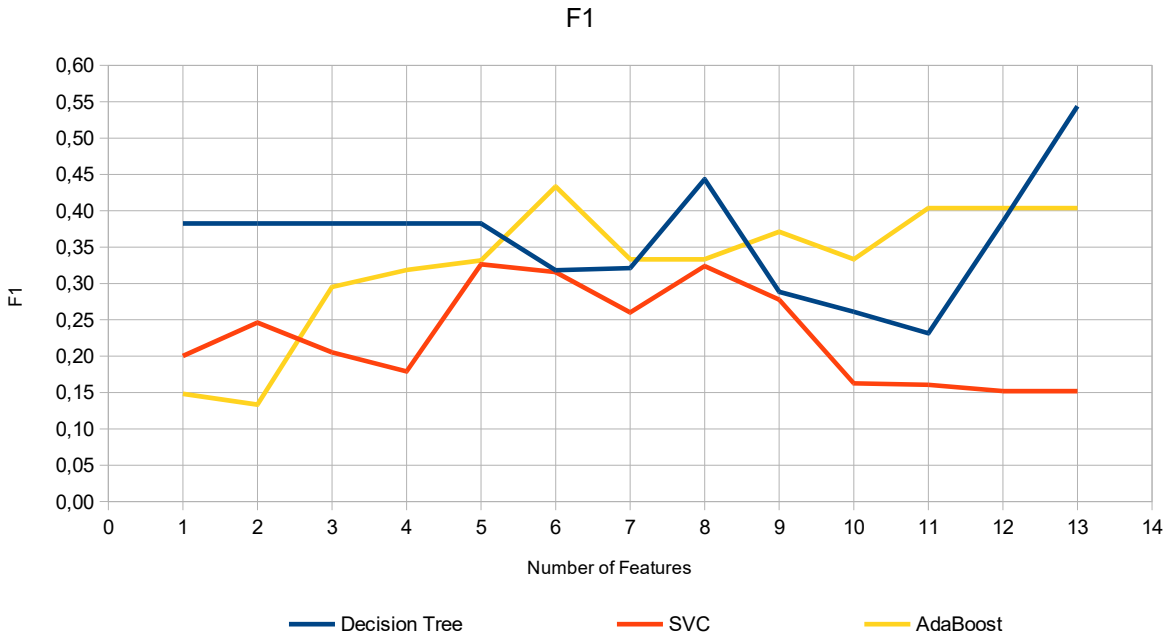

Figura 5 – Grafico F1 score – desempenho do algoritmo com variacao dos recursos utilizados

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"]

Neste projeto foram utilizados três diferentes algoritmos: Decision Tree, SVC e AdaBoost.

Tabela 4 – Desempenho dos modelos

	Number of Features	Accuracy	Precision	Recall	F1	F2
Decision Tree	13	0,78342	0,31068	0,70780	0,43182	0,56370
SVC	5	0,72991	0,23395	0,58160	0,33368	0,44835
AdaBoost	6	0,85007	0,33825	0,30260	0,31943	0,30912

Numa análise comparativa de desempenho, o algoritmo Decision Tree obteve os melhores resultados, portanto, foi o algoritmo escolhido.

Os resultados com os novos recursos no modelo e sem esses recursos são apresentados na Tabela 5, abaixo.

Tabela 5 – Desempenho dos modelos com e sem os novos recursos

	Novas Features	Accuracy	Precision	Recall	F1	F2
Decision Tree	**com**	0,78342	0,31068	0,70780	0,43182	0,56370
	sem	0,82114	0,24855	0,26600	0,25698	0,26232
SVC	**com**	0,72991	0,23395	0,58160	0,33368	0,44835
	sem	0,73188	0,23283	0,56900	0,33045	0,44151
AdaBoost	**com**	0,85007	0,33825	0,30260	0,31943	0,30912
	sem	0,69149	0,14992	0,35400	0,21064	0,27825

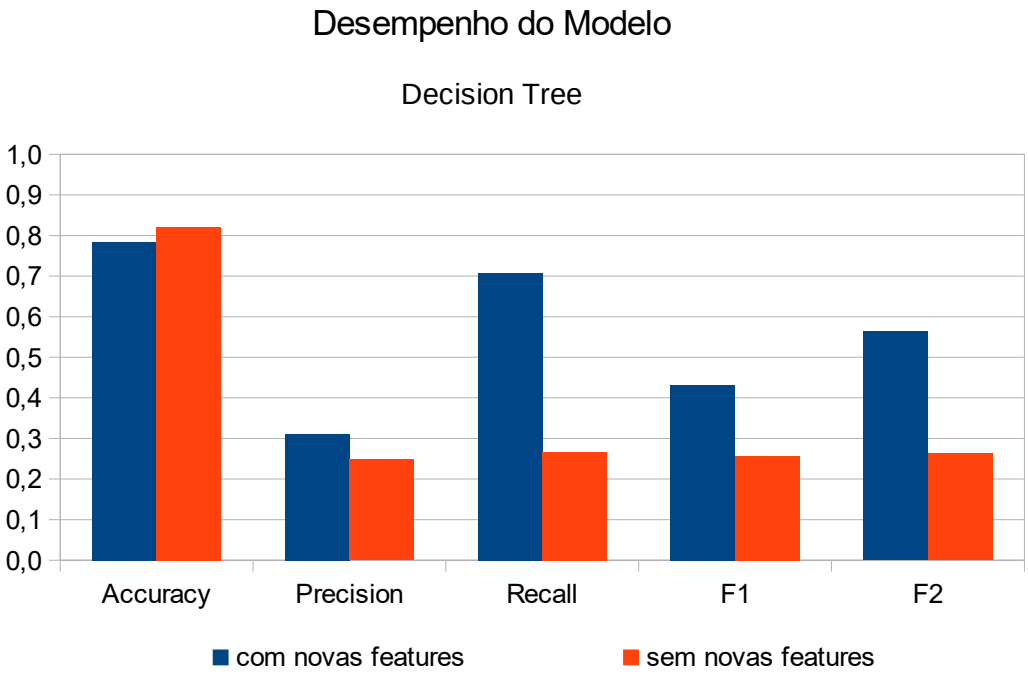

Figura 6 – Desempenho do modelo Decision Tree (com e sem os novos recursos)

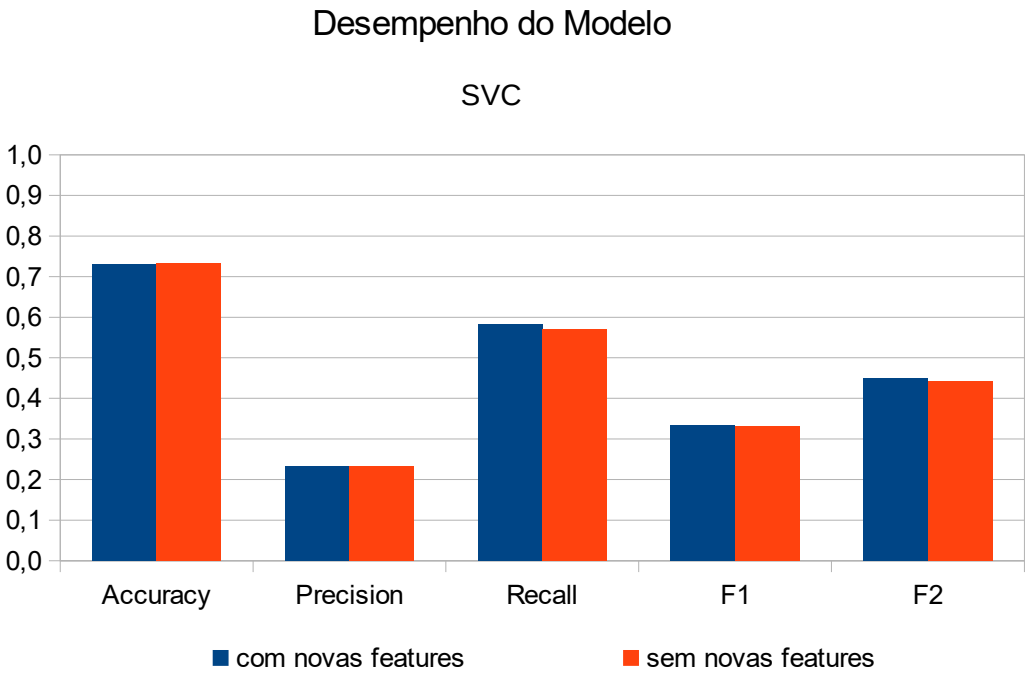

Figura 7 – Desempenho do modelo SVC (com e sem os novos recursos)

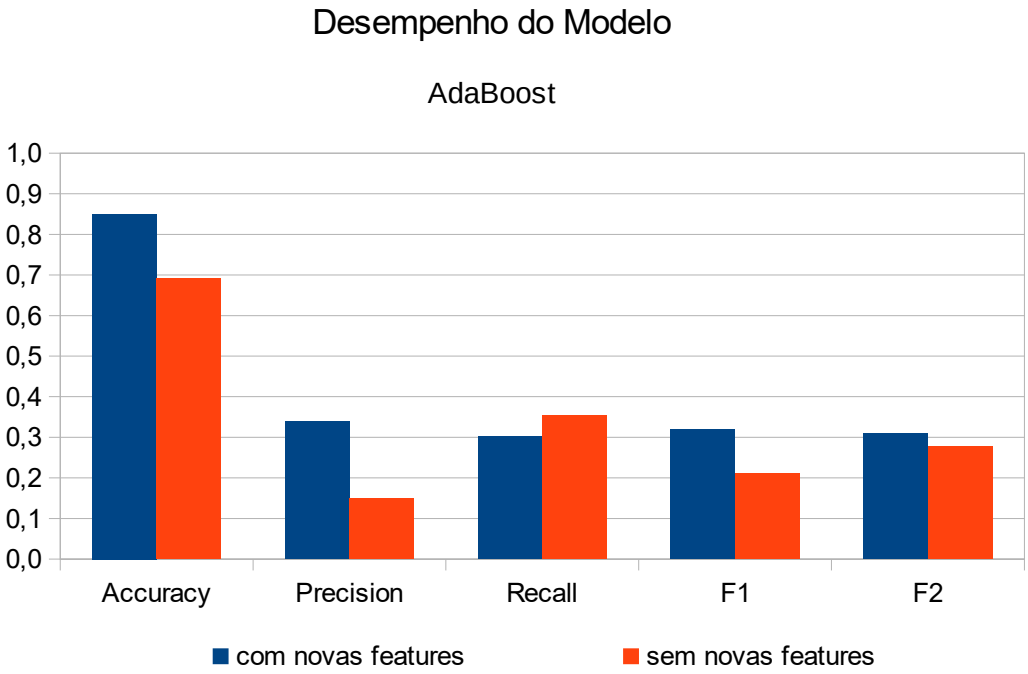

Figura 8 – Desempenho do modelo Adaboost (com e sem os novos recursos)

4. O que significa afinação (tuning) dos parâmetros de um algoritmo e o que pode acontecer se você não fizer isso bem? Como você afinou 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"]

Afinação ou tuning consiste em encontrar a melhor combinação entre as possibilidades de parâmetros. Uma vez que esses parâmetros podem influenciar o desempenho desses modelos, torna-se essencial utilizar uma função capaz de encontrar a melhor combinação desses parâmetros. Neste projeto, o tuning foi realizado com o auxílio da função GridSearchCV que é uma forma de analisar sistematicamente múltiplas combinações de parâmetros, fazendo validação cruzada ao longo do processo, para determinar qual calibragem (parametrização) fornece o melhor desempenho. Para os algoritmos testados (Decision Tree, SVC e AdaBoost) foram encontrados os seguintes parâmetros:

DecisionTreeClassifier(class_weight='balanced', criterion='gini', max_depth=2,
max_features='auto', max_leaf_nodes=2,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=1.0,
min_weight_fraction_leaf=0.0, presort=False, random_state=42,
splitter='best')

SVC(C=0.05, cache_size=200, class_weight='balanced', coef0=0.0,
decision_function_shape='ovo', degree=3, gamma='auto', kernel='linear',
max_iter=1000, probability=False, random_state=42, shrinking=False,
tol=0.001, verbose=False)

AdaBoostClassifier(algorithm='SAMME', base_estimator=None, learning_rate=1.5,
n_estimators=10, random_state=42)

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"]

A validação é processo para verificar a performance do modelo com a utilização de dados não utilizados durante a etapa de treinamento. Um erro muito comum nesta etapa é o *overfitting* em que o modelo funciona muito bem para prever os dados de treinamento, porém tem uma performance muito inferior quando utilizado com dados diferentes.

Uma das formas de evitar o *overfitting* consite em realizar a validação cruzada, um processo aleatório em que o conjunto de dados disponíveis é separado em um conjunto para o treinamento do modelo e outro conjunto para a etapa de validação (ou teste).

Neste projeto, na etapa de validação, devido ao fato das classes possuírem tamanhos bem diferentes (POI: 18/143, não-POI:125/143) e também pelo fato dos dados estarem organizados em ordem alfabética, os dados precisam ser embaralhados e estratificados (preservação da porcentagem de amostras para cada classe) antes de serem divididos nos conjuntos de treinamento e de teste. Desse modo, a função StratifiedShuffleSplit foi adotada em relação à função Kfold, visto que essa última não embaralha os dados e nem garante a estratificação.

Assim, neste projeto, a função StratifiedShuffleSplit foi adotada e parametro "test_size" ajustado em 0.3, ou seja, 30% dos dados separados para o conjunto de teste.

6. Dê pelo menos duas métricas de avaliação e seu desempenho médio para cada uma delas. Explique uma interpretação de suas métricas no contexto da tarefa sobre o desempenho do seu algoritmo. [rubrica relevante: "uso de métricas de avaliação"]

Para tarefas de classificação, define-se:

- **Precision (precisão ou valor preditivo positivo):** é a fração de pessoas rotuladas que são POI, ou seja, é a relação entre o número de pessoas rotuladas corretamente como POI (verdadeiros positivos) e o número total de pessoas rotuladas como POI, incluindo as pessoas incorretamente rotulados como POI (verdadeiros positivos e falsos positivos).
- **Recall (revocação ou sensibilidade):** é a fração de POI que são corretamente rotulados, ou seja, indica a relação entre o número de pessoas rotuladas corretamente como POI (verdadeiros positivos) e o número total de POI, incluindo as pessoas que não foram rotuladas como POI e que deveriam ter sido (verdadeiros positivos e falsos negativos).

Essas duas medidas são por vezes utilizadas em conjunto no F1 Score (ou f-measure) para fornecer uma única medição para um sistema. O F1 Score, por ser uma média harmônica entre precision e recall, está muito mais próxima do menor valor do que uma média aritmética simples, ou seja, um F1 score baixo indica que ou precision ou recall está baixo.

Na tarefa de classificação, a precisão se refere ao número de falsos positivos, ou seja, para um modelo de alta precisão, cada pessoa rotulada como POI, de fato, é POI (mas não diz nada sobre o número de pessoas que são POI e que não foram rotuladas corretamente). Por sua vez, o recall se refere ao número de falsos negativos, ou seja, para um modelo de alto recall, significa que cada POI foi rotulado como POI (mas não diz nada sobre o número de pessoas que foram incorretamente rotuladas como POI).

Referências

Slack – UDACITY-BR

https://scikit-learn.org/stable/modules/generated/klearn.feature_selection.SelectKBest.html

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html

https://scikit-learn.org/stable/tutorial/statistical_inference/model_selection.html

https://pt.wikipedia.org/wiki/Precis%C3%A3o_e_revoca%C3%A7%C3%A3o

https://gabrielschade.github.io/2019/03/12/ml-classificacao-metricas.html

https://towardsdatascience.com/machine-learning-workflow-on-diabetes-data-part-02-11262b7f7a5c