Fundamentos de Data Science II

Projeto 5 – Introdução a Machine Learning

Evandro Silva

Identificar fraude no Email da Enron

Sumário

Introdução

O Intuito deste projeto é colocar em prática as lições aprendidas no módulo de Introdução a Machine Learning do Nanodegree Data Science Foundations II da Udacity.

Objetivo

Neste projeto, você irá atuar como um detetive, e usar suas habilidades em aprendizagem de máquina para criar um algoritmo que identifique os funcionários da Enron que podem ter cometido fraude baseando-se no conjunto de dados público entitulado "Enron financial and email".

Questões

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”]**

Resposta

Trabalhar como detective em um conjunto de dados da base de email da Eron, a fim de identificar quais funcionários são possíveis fraudadores. Existema algumas informações que facilitam o trabalho, como os salários, bônus, os pagamentos recebidos e realizados, a quantidade de e-mail que cada funcionário enviou e recebeu, etc. Porém, muitos registros estão com informações em branco.Com base nas analises do conjunto de 146, Três registros foram descartados, pois suas informações seria incorretas para a analise. Dois graficos Scater foram criados com base no Sálario e o Bonus dos funcionários com e sem os outliers.

```
Quantidade total de registros: 146
Quantidade total de features: 21
Quantidade de POI's: 18
Quantidade de não POI's: 128
```

```
print "Quantidade total de registros Com outliers: {}\n".format(len(data_dict_woo))

data_dict_woo.pop('TOTAL', None) #Não é um funcionário
data_dict_woo.pop('LOCKHART EUGENE E', None) #Não é um funcionário
data_dict_woo.pop('THE TRAVEL AGENCY IN THE PARK', None) #Não é um funcionário

print "Quantidade total de registros Sem outliers: {}\n".format(len(data_dict_woo))
```

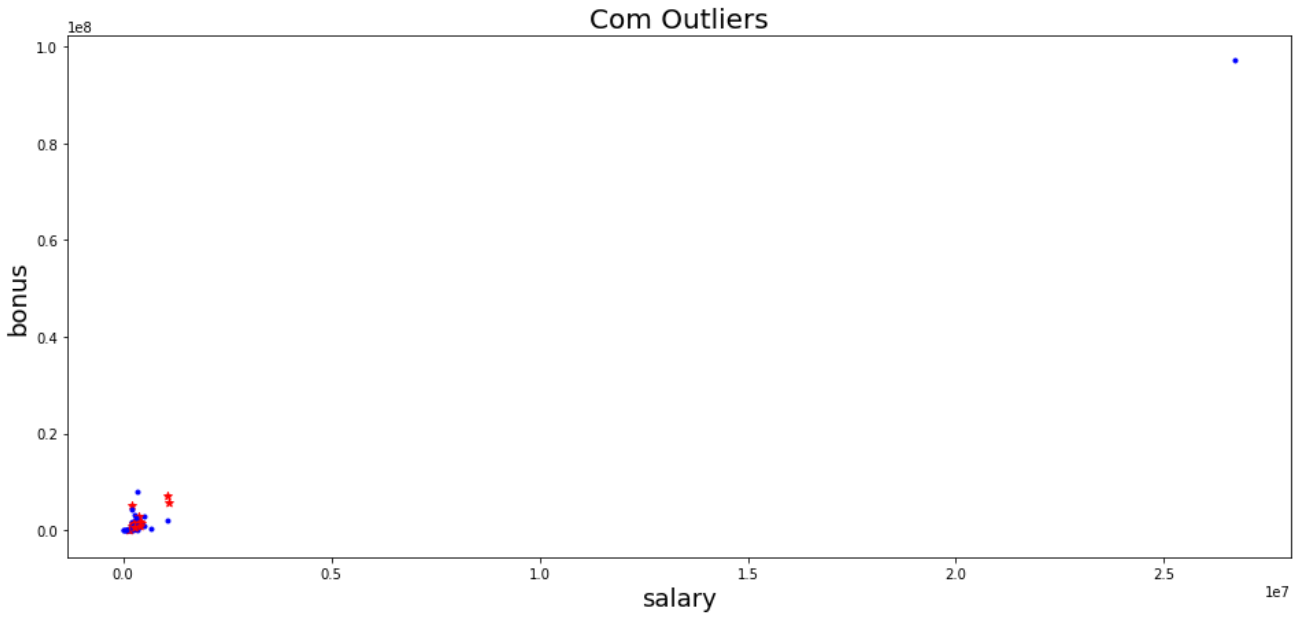
```
Quantidade total de registros Com outliers: 146

Quantidade total de registros Sem outliers: 143
```

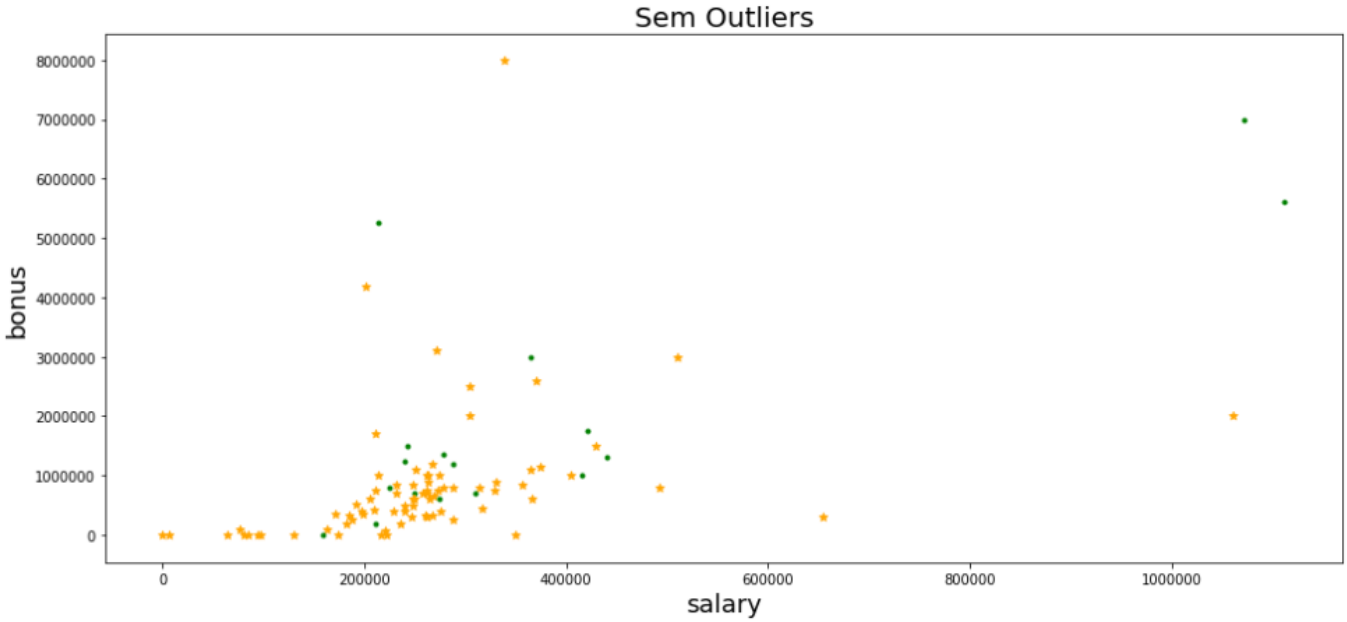
	count	unique	top	freq
salary	146	95	NaN	51
to_messages	146	87	NaN	60
deferral_payments	146	40	NaN	107
total_payments	146	126	NaN	21
exercised_stock_options	146	102	NaN	44
bonus	146	42	NaN	64
restricted_stock	146	98	NaN	36
shared_receipt_with_poi	146	84	NaN	60
restricted_stock_deferred	146	19	NaN	128
total_stock_value	146	125	NaN	20
expenses	146	95	NaN	51
loan_advances	146	5	NaN	142
from_messages	146	65	NaN	60
other	146	93	NaN	53

Como forma de visualizar os dados analisados, foi criado um gráfico de scatter de Bonus X Salário.

Porém, a visualização ficou comprometida pela Outliers.

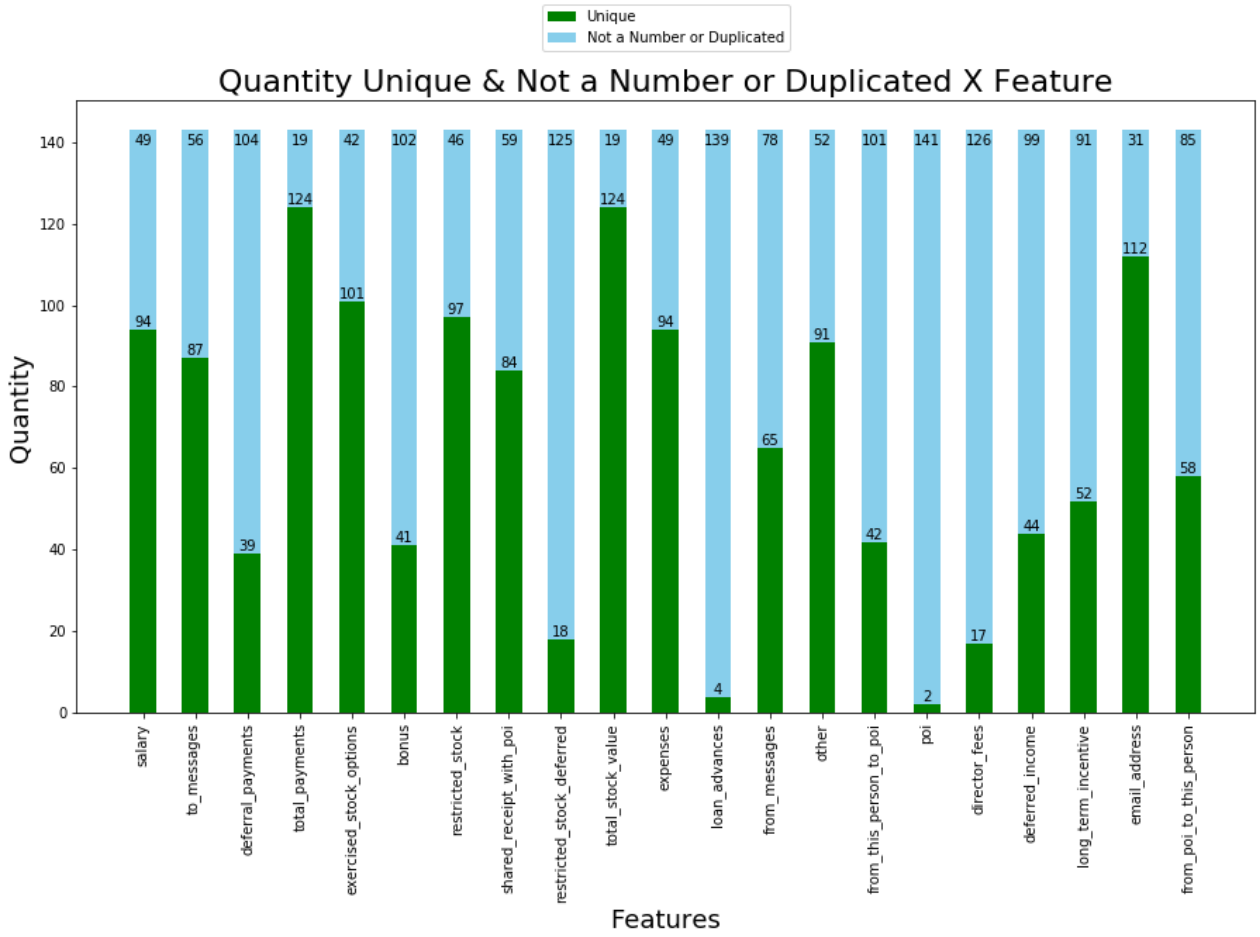


Após remover os Outliers, obtivemos um gráfico mais condizente com os dados que representam melhor a amostra.



O Gráfico abaixo representa uma análise de cada uma das Features utilizada.

Referente a Valores únicos, Não atribuídos e Duplicados.



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”]

Resposta

Como haviam alguns outliers que poderiam distorcer o resultado, foi utilizado o escalonamento de recursos e eles foram redimensionados usando sklearn min/max scaler. Um dado adicional, é que as características foram escolhidas manualmente.

O algoritmo SelectKbest foi utilizado com as features para obter as melhores pontuações de Recall e Precison durante os testes. Com o uso da StratifiedShuffleSplit foi criado o conjunto de testes. Baseado na lista de POIs, as features abaixo foram as classificadas como atributibutos para os testes.

```
features: salary
features: bonus
features: deferred_income
features: total_payments
features: loan_advances
features: total_stock_value
features: exercised_stock_options
features: long_term_incentive
features: shared_receipt_with_poi
features: restricted_stock
```

As features fraction_from_poi_email e fraction_to_poi_email, foram criadas e adicionadas ao conjunto de features.

Como o e-mail provavelmente foi utilizado para troca de informações entre os suspeitos de fralda, calcular a fração de e-mails enviados por POI, poderia ser uma boa maneira de identificar os suspeitos de terem realizado as fraudes.

Conforme exemplo abaixo:

```
DIETRICH JANET R
- fraction_from_poi_email = 0.1186 (305 / 2572)
- fraction_to_poi_email = 0.2222 (14 / 63)
```

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”]

Foi utilizado SelectKBest para seleção dos recursos, a fim de obter as melhores pontuações de Precisão e Recall durante vários testes.

```
features: salary score: 15.858731
features: bonus score: 30.728775
features: deferred_income score: 8.792204
features: total_payments score: 8.959137
features: loan_advances score: 7.037933
features: total_stock_value score: 10.633852
features: exercised_stock_options score: 9.680041
features: long_term_incentive score: 7.555120
features: shared_receipt_with_poi score: 10.722571
features: restricted_stock score: 8.058306
```

	Algorithms	New Features	Accuracy
0	GaussianNB	N	0.88372
1	KNeighborsClassifier	N	0.90698
2	SVC	N	0.88372
3	AdaBoostClassifier	N	0.81395
4	GaussianNB	S	0.88372
5	KNeighborsClassifier	S	0.90698
6	SVC	S	0.88372
7	AdaBoostClassifier	S	0.95349

Com a inclusão das novas features, accuracy do AdaBoosterClassifier atingiu o melhor resultado dos 4 algoritmos testados.

Após identificar que o algoritmo AdaBoosterClassifier tinha um melhor performance para o dados analisados, fiz mais alguns testes utilizando SelectKBest de 1 a 10 para Features Default e Novas Features. Eles apresentaram valores parecidos dentro de dos respectivos grupos de Features(Default ou Novas).

Porém, para este teste ficou evidente que o SelectKBest com valor k=4 foi mais assertivo. Outro ponto, é que a precisão do grupo Features Nova quase dobrou em relação ao grupo de Fature Default. Os valores de Falso posi tivo e Falso negativo foi menor.

Abaixo uma tabela com mais detalhes e no rodapé da tabela a configuração dos parâmetros utilizado para otimi zar Algoritmo:

kbest	New Features	Accuracy	Precision	Recall	F1	F2	Total predictions	True positives	False positives	False negatives	True negative	time	unidade(t)
1	Y	0.91247	0.70313	0.59450	0.64427	0.61346	15000	1189	502	811	12498	76.154	s
2	Y	0.91260	0.70397	0.59450	0.64462	0.61358	15000	1189	500	811	12500	76.127	s
3	Y	0.91260	0.70373	0.59500	0.64481	0.61397	15000	1190	501	810	12499	76.195	s
4	Y	0.91267	0.70414	0.59500	0.64499	0.61404	15000	1190	500	810	12500	76.254	s
5	Y	0.91260	0.70397	0.59450	0.64462	0.61358	15000	1189	500	811	12500	76.098	s
6	Y	0.91267	0.70390	0.59550	0.64518	0.61442	15000	1191	501	809	12499	76.012	s
7	Y	0.91253	0.70355	0.59450	0.64444	0.61352	15000	1189	501	811	12499	76.061	s
8	Y	0.91260	0.70348	0.59550	0.64500	0.61436	15000	1191	502	809	12498	76.268	s
9	Y	0.91260	0.70373	0.59500	0.64481	0.61397	15000	1190	501	810	12499	76.313	s
10	Y	0.91253	0.70355	0.59450	0.64444	0.61352	15000	1189	501	811	12499	76.245	s
1	N	0.83180	0.35888	0.33250	0.34519	0.33746	15000	665	1188	1335	11812	75.729	s
2	N	0.83167	0.35864	0.33300	0.34535	0.33783	15000	666	1191	1334	11809	75.607	s
3	N	0.83167	0.35849	0.33250	0.34501	0.33739	15000	665	1190	1335	11810	75.715	s
4	N	0.83193	0.35942	0.33300	0.34570	0.33797	15000	666	1187	1334	11813	76.130	s
5	N	0.83173	0.35868	0.33250	0.34510	0.33743	15000	665	1189	1335	11811	76.120	s
6	N	0.83187	0.35892	0.33200	0.34494	0.33706	15000	664	1186	1336	11814	75.786	s
7	N	0.83167	0.35849	0.33250	0.34501	0.33739	15000	665	1190	1335	11810	75.821	s
8	N	0.83180	0.35903	0.33300	0.34553	0.33790	15000	666	1189	1334	11811	76.190	s
9	N	0.83187	0.35922	0.33300	0.34561	0.33793	15000	666	1188	1334	11812	75.957	s
10	N	0.83180	0.35888	0.33250	0.34519	0.33746	15000	665	1188	1335	11812	75.754	s

```
AdaBoostClassifier(algorithm='SAMME.R',
      base_estimator=DecisionTreeClassifier(class_weight='balanced', criterion='gini', max_depth=1,
       max_features=None, max_leaf_nodes=None,
       min_impurity_split=1e-07, min_samples_leaf=3,
       min_samples_split=2, min_weight_fraction_leaf=0.0,
       presort=False, random_state=None, splitter='best'),
      learning_rate=0.95, n_estimators=50, random_state=None)
```

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"]

Ao atualizar os parâmetros de um classificador, estamos fazendo ajustes para otimizar o mesmo a fim de obter a melhor combinação entre desempenho e assertividade na classificação.
Neste projeto e estudos, pode-se perceber que vários algoritmos em Machine Learning possuem valores para atributos padrões.
Alguns destes padrões já são as melhores combinações e altera-los, pode ter o efeito contrário. Porém, acho int eressante fazer testes com os ajustes para confirmar que o padrão é o melhor para o modelo de dados.

Para chegar na melhor configuração que eu consegui identificar para o Algoritmo que eu escolhi, utilizei a funçã o GridSearchCV para determinar automaticamente os parâmetros otimizados.
Ao receber um conjunto de parâmetros, essa função avalia todas as combinações possíveis e retorna um classifi cador que fornece a melhor pontuação/combinação.

Algorithmn	Accuracy	Precision	Recall
RandomForest	0.86527	0.48730	0.20150
Descisiontree	0.69307	0.23055	0.55700
AdaBoost	0.91267	0.70373	0.59500

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 a previsão do seu modelo em relação a dados que não foram usados para tre inar seu classificador/algoritmo, ou seja, separar parte dos seus dados em conjuntos de treinamentos.
Com isso, você consegue enviar uma parte para o classificador e outra para avaliar a performance do classificad or.

O processo chamado overfitting, ocorre quando você treina o algoritmo em todos os conjuntos de dados dispon íveis, este á um erro clássico. O correto, seria dividir em dados de treinamento e teste.

No projeto, foi utilizado o StratifiedShuffleSplit para criar vários conjuntos de dados aleatórios para o modelo fin al.

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"]

Accuracy(acurácia) e precision(precisão), foram duas das métricas utilizadas para avaliar o modelo de previsão d e POIs.
Acurácia é o quão próximo uma medida é do valor final verdadeiro.
Precisão é quantos itens selecionados foram identificados como relevantes.

Algorithmn	Accuracy	Precision	Recall
AdaBoost	0.91267	0.70373	0.59500