1- A Enron foi uma empresa de distribuição de energia e gás e chegou a ser a sétima maior empresa do Estados Unidos, mas após várias fraudes na sua corporação a empresa entrou em colapso. O governo americano abriu dezenas de investigações e identificou que a empresa havia manipulado os seus balanços, caracterizando assim fraude contábil, assim, muitos dados que normalmente são confidenciais, tornaram-se públicos. Dentre eles um relatório com dados financeiros de funcionários e membros do conselho e também milhares de arquivos de e-mails.

O objetivo deste projeto é utilizar técnicas de aprendizado de máquina (machine learning) que consiga aprender com os dados disponibilizados e realizar predições indicando se o funcionário é ou não um POI (Person of interest—Pessoa de interesse).

O conjunto de dados disponível possue 20 features (atributos) do tipo financeiros e do tipo e-mail, e um label se é POI ou não. Como features iniciais para a análise foram utilizadas todas do tipo financeiro e na de e-mail só a feature 'email_address' não foi utilizada, por ser uma string que para essa finalidade não possui uma informação relevante.

A análise inicial mostra que existe 146 amostras, sendo 18 POIs e 128 não-POIs. A próxima análise feita é para retornar amostras com muitos valores faltantes. Para este caso, 3 amostras foram retornadas:

WODRASKA JOHN{

'salary':'NaN',

'to_messages':'NaN',

'deferral_payments':'NaN',

'total_payments':189583,

'loan_advances':'NaN',

'bonus':'NaN',

'email_address':'john.wodraska@enron.com',

'restricted_stock_deferred':'NaN',

'deferred_income':'NaN',

'total_stock_value':'NaN',

'expenses':'NaN',

'from_poi_to_this_person':'NaN',

'exercised_stock_options':'NaN',
'from_messages':'NaN',
'other':189583,
'from_this_person_to_poi':'NaN',
'poi':False,
'long_term_incentive':'NaN',
'shared_receipt_with_poi':'NaN',
'restricted_stock':'NaN',
'director_fees':'NaN'
}

LOCKHART EUGENE E{
'salary':'NaN',
'to_messages':'NaN',
'deferral_payments':'NaN',
'total_payments':'NaN',
'loan_advances':'NaN',
'bonus':'NaN',
'email_address':'NaN',
'restricted_stock_deferred':'NaN',
'deferred_income':'NaN',
'total_stock_value':'NaN',
'expenses':'NaN',
'from_poi_to_this_person':'NaN',
'exercised_stock_options':'NaN',
'from_messages':'NaN',
'other':'NaN',
'from_this_person_to_poi':'NaN',
'poi':False,
'long_term_incentive':'NaN',
'shared_receipt_with_poi':'NaN',
'restricted_stock':'NaN',
'director_fees':'NaN'
}

THE TRAVEL AGENCY IN THE PARK'{
'salary':'NaN',
'to_messages':'NaN',
'deferral_payments':'NaN',
'total_payments':362096,
'loan_advances':'NaN',
'bonus':'NaN',
'email_address':'NaN',
'restricted_stock_deferred':'NaN',
'deferred_income':'NaN',
'total_stock_value':'NaN',
'expenses':'NaN',
'from_poi_to_this_person':'NaN',
'exercised_stock_options':'NaN',
'from_messages':'NaN',
'other':362096,
'from_this_person_to_poi':'NaN',
'poi':False,
'long_term_incentive':'NaN',
'shared_receipt_with_poi':'NaN',
'restricted_stock':'NaN',
'director_fees':'NaN'
}

Já fazendo uma análise visual nos dados, é possível notar que o valor de uma amostra está bem desproporcional dos valores das demais, tanto no salário como no bônus, como mostra as Figuras 1 e 2.

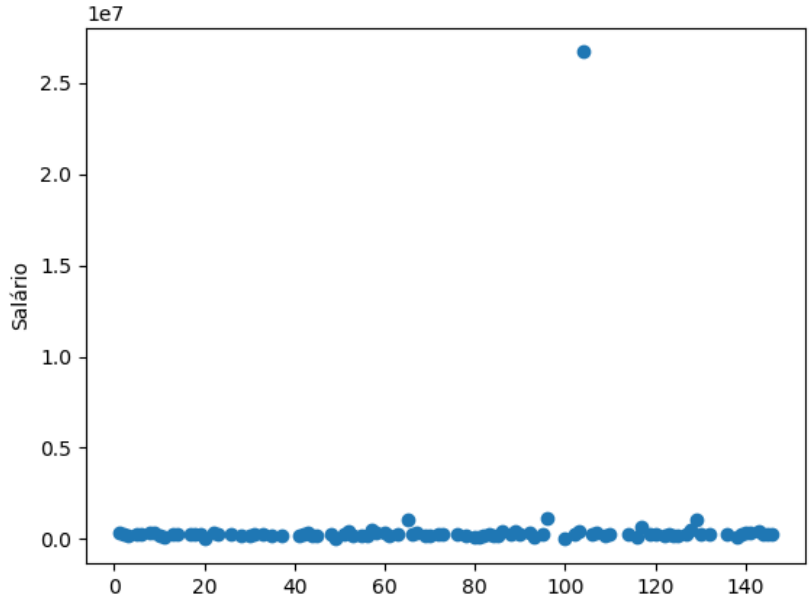

Figura 1- Salários dos amostras do conjunto

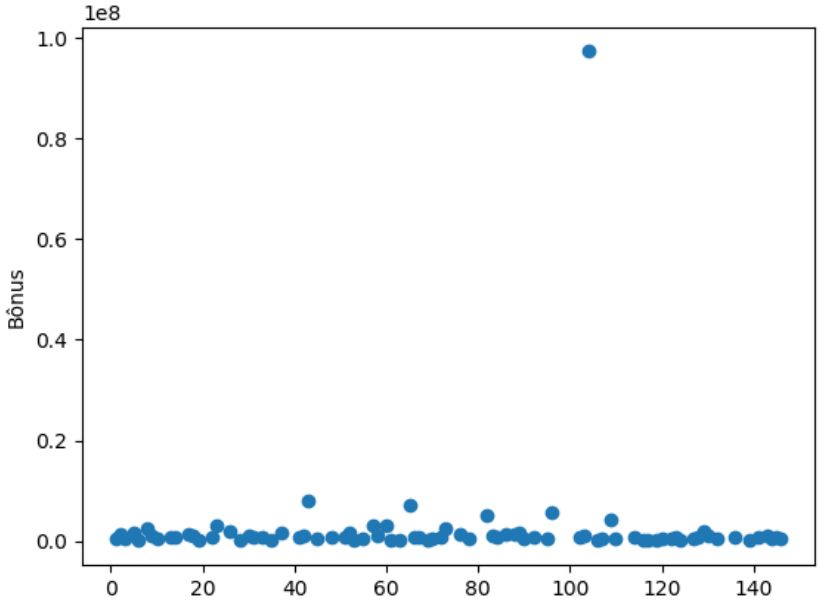

Figura 2-- Bônus dos amostras do conjunto

Essa amostra é ‘TOTAL’ e ela não se trata da informação de um funcionário, e sim de uma linha contendo o valor total de todas as colunas. Portanto é um Outliers e será removido junto com as três amostras anteriores.

2 – Foram criadas 3 novas features:

fraction_from_poi: percentual de mensagens recebidas de um POI em relação ao total recebida.

fraction_to_poi: percentual de mensagens enviadas para um POI em relação ao total enviada.

fraction_total_stock_value: percentual de ações da amostra em relação ao total de ações.

Assim, ficou um total de 23 features no conjunto de dados. Foi utilizado o SelectKBest, para selecionar a metade das features mais significativas (k=12). A seguir estão as 12 principais features e seus respectivos scores:

Feature : exercised_stock_options,

Score: 24.532722463057976

Feature: 'fraction_total_stock_value',

Score: 23.898259813869437

Feature: 'total_stock_value',

Score:23.898259813869416

Feature: 'bonus',

Score:20.524645181851792

Feature: 'salary',

Score:18.003739993113935

Feature: 'fraction_to_poi',

Score:16.17789144634484

Feature: 'deferred_income',

Score:11.321486775141238

Feature: 'long_term_incentive',

Score:9.772103538408254

Feature: 'restricted_stock',

Score:9.07907666167087

Feature: 'total_payments',

Score:8.675460131394738

Feature: 'shared_receipt_with_poi',

Score:8.432635423024681

Feature: 'loan_advances',

Score:7.1253824688830685

Duas das três featuras criadas, fraction_to_poi e fraction_total_stock_value, estão entre as 12 principais do conjunto, sendo que fraction_total_stock_value ficou em segundo.

3 – Os dois algoritmos implementados foram RandomForestClassifier e DecisionTreeClassifier, sendo que o primeiro teve o melhor desempenho e foi implementado no modelo final.

4 – Ajustar os parâmetros de um algoritmo permite que se obtenha a melhor performance do mesmo. Para os dois algoritmos utilizados neste projeto, foi implementado o GridSearchCV para a escolha dos me lhores parâmetros. No RandomForestClassifier os parâmetros ajustados foram:

```
parameters = {
        "criterion": ['entropy', 'gini'],
        "n_estimators": [25, 50, 75],
       "bootstrap": [False, True],
        "max_depth": [3, 5, 10],
        "max_features": ['auto', 0.1, 0.2]
         }
```

Sendo que os melhores parâmetros foram:

{'bootstrap': False, 'criterion': 'entropy', 'max_features': 0.2, 'n_estimators': 50}

Já em DecisionTreeClassifier o GridSearch foi realizado com os seguintes parâmentros:

parameters={'min_samples_split' : range(10,500,20),

"criterion": ["gini", "entropy"]}

Sendo que os selecionados foram:

{'criterion': 'gini', 'min_samples_split': 90}

5 - A validação é importante por permitir a análise se o treinamento do método utilizado foi eficaz e, assim, permite que se utilize o método com uma base de dados independente.

Na validação divide-se o conjunto de dados em treinamento e teste. Para o projeto, foi utilizado o método train_test_split, que divide os dados em treinamento e teste, sendo utilizado, 85% para treinamento e 15% para teste.

6 – As métricas de avaliação utilizadas foram Precision, Recall e Accuracy. A métrica Precision é definida por:

$$\frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP) + Falsos\ Positivos\ (FP)}$$

Que significa que o valor do Precision é o quanto que é confiável na predição quando o algoritmo afirma que o resultado é um determinado valor.

A métrica Recall é definida por:

$$\frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP) + Falsos\ Negativos\ (FN)}$$

Que significa que se o valor do Recall for alto, as predições das classes serão feitas de forma exatas. Já a Accuracy é definida por:

$$\frac{Verdadeiros\ Positivos\ (TP) + Verdadeiros\ Negativos\ (TN)}{Verdadeiros\ Positivos\ (TP) + Verdadeiros\ Negativos\ (TN) + Falsos\ Negativos\ (FN) + Falsos\ Positivos\ (FP)}$$

Que seria uma porcentagem de quanto o classificador está correto.

Neste projeto, os classificadores forem testado com o conjunto de dados completo e posteriormente só com as 12 melhores features. Para o primeiro caso, a avaliação obtida foi:

	Accuracy	Prediction	Recall
RandomForestClassifier	0.860465116279	0.5	0.166666666667
DecisionTreeClassifier	0.813953488372	0.25	0.166666666667

Já com o conjunto de 12 features, obteve-se:

	Accuracy	Prediction	Recall
RandomForestClassifier	0. 863636363636	0.5	0.33
DecisionTreeClassifier	0.720930232558	0.2	0.33

Sendo assim, o algoritmo de mechine learning final é o RandomForestClassifier.