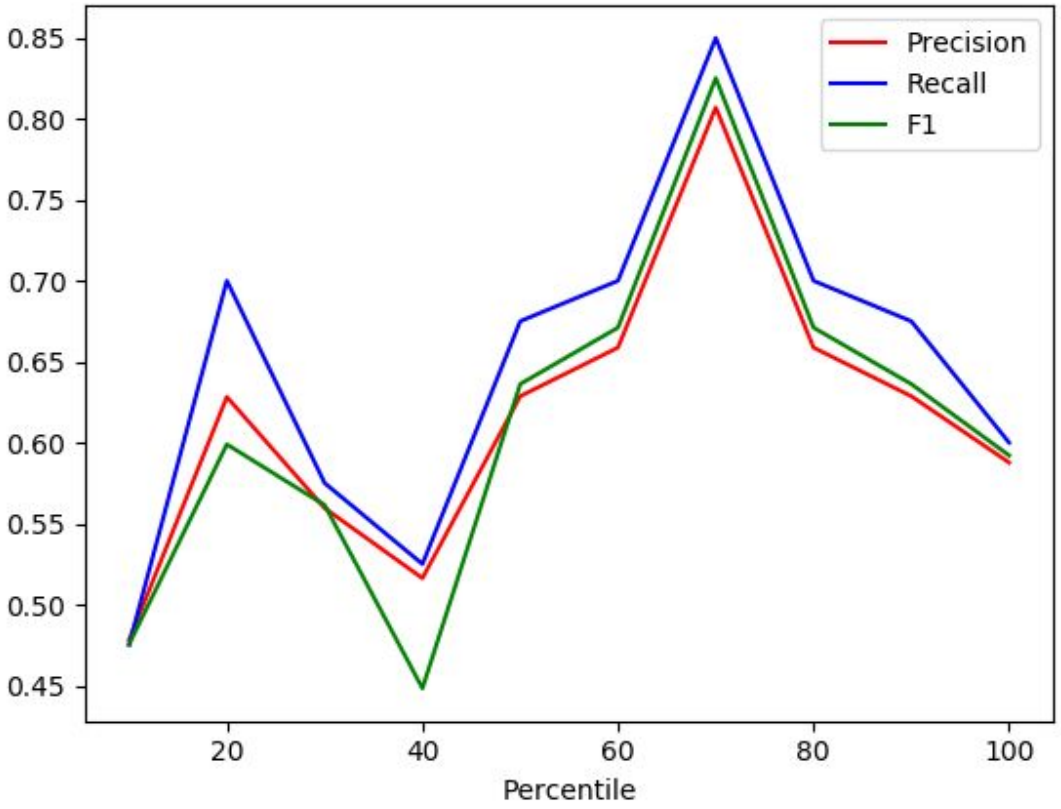
O objetivo desse projeto foi usar o conjunto de dados com informações financeiras e de emails de funcionários da *Enrom Corporation* para identificar pessoas de interesse (*POI*) que no caso são funcionários com participação na fraude que ficou famosa no início dos anos 2000. Existem dados de 146 funcionários sendo 18 deles *POIs* e 128 *non-POIs*. A utilização de *machine learning* ajuda na detecção de padrões e/ou similaridades não triviais entre os funcionários.

É interessante citar que existe uma quantidade considerável de funcionários que não possuem alguma das informações como também existem alguns casos de funcionário com valores fora do esperado ou extremamente diferente dos demais (outliers). Esses valores foram removidos do conjunto de dados resultando na remoção de 63 *non-POIs. São eles:*

CORDES WILLIAM R	MCCARTY DANNY J	MORAN MICHAEL P	FOWLER PEGGY
MEYER ROCKFORD G	BERBERIAN DAVID	REDMOND BRIAN L	CHRISTODOULOU DIOMEDES
HORTON STANLEY C	WAKEHAM JOHN	BAZELIDES PHILIP J	JAEDICKE ROBERT
HUMPHREY GENE E	POWERS WILLIAM	THORN TERENCE H	WINOKUR JR. HERBERT S
GIBBS DANA R	BANNANTINE JAMES M	FOY JOE	BROWN MICHAEL
LOWRY CHARLES P	DUNCAN JOHN H	LOCKHART EUGENE E	BADUM JAMES P
WESTFAHL RICHARD K	LEMAISTRE CHARLES	OVERDYKE JR JERE C	HUGHES JAMES A
WALTERS GARETH W	KISHKILL JOSEPH G	PEREIRA PAULO V. FERRAZ	BHATNAGAR SANJAY
CHAN RONNIE	PIRO JIM	BLAKE JR. NORMAN P	YEAP SOON
BELFER ROBERT	WROBEL BRUCE	SHERRICK JEFFREY B	HAYSLETT RODERICK J
WODRASKA JOHN	MEYER JEROME J	PRENTICE JAMES	FUGH JOHN L
URQUHART JOHN A	MCDONALD REBECCA	GRAY RODNEY	SAVAGE FRANK
WHALEY DAVID A	SCRIMSHAW MATTHEW	THE TRAVEL AGENCY IN THE PARK	IZZO LAWRENCE L
HAUG DAVID L	GATHMANN WILLIAM D	NOLES JAMES L	MARTIN AMANDA K
MENDELSOHN JOHN	GILLIS JOHN	TOTAL	GRAMM WENDY L
CLINE KENNETH W	LEWIS RICHARD	HAYES ROBERT E	

A escolha das features foi através de validação cruzada do percentil de features que deveria ser usado. Foram testados os múltiplos de 10 até alcançar 100%. No gráfico abaixo é possível visualizar variação das métricas com a variação da quantidade de features utilizadas.



A solução final utiliza 15 *features* (70%) com destaque para *bonus*, *expenses* e *other* que obtiveram maior importância para o modelo. Uma das caracteristicas de florestas aleatórias é a possibilidade de calcular a importância de cada variável na criação do modelo.

- bonus --> importância: 0.19450184
- expenses --> importância: 0.15975696
- other --> importância: 0.10185397
- o restante pode ser visto no código

O resultado final não utilizou escalonamento das features, entretanto, é interessante citar que tal etapa se fez necessária para o teste do algoritmo kNN.

Como comentado, algumas das features tentam expressar, de forma absoluta, o quanto um funcionário interagiu com um POI. Achei que seria mais interessante que isso foi feito levando

em consideração o quanto o funcionário se comunicava em geral. Por isso criei as variáveis: "*from_poi_to_this_person_ratio*", "from_this_person_to_poi_ratio" e "shared_receipt_with_poi_ratio". A solução final utiliza apenas *shared_receipt_with_poi_ratio.*

Outro ponto importante é que por causa da falta de balanceamento entre as classes, foi aplicada uma etapa de *oversampling* com o objetivo de aumentar a quantidade de registros da classe minoritária, no caso POIs. Esse passo foi muito importante principalmente para melhor o *recall* do resultado.

A solução final usou florestas aleatórias como algoritmo de classificação. Entretanto, até chegar no resultado final foram testados os algoritmos: *naive bayes*, kNN, SVC e *decision trees*. A utilização de SVC estranhamente obteve os piores resultados junto com kNN e naive bayes com dificuldade de ultrapassar o limiar de 0.2 para a precisão. *Decision tree* obteve resultados melhores mas não tão bons quando florestas aleatórias.

Uma parte importante do processo foi o ajuste dos parâmetros do algoritmos (hiperparâmetros). Nessa etapa, é possível configurar como o modelo irá aprender os dados de forma que o resultado final seja útil para casos que não estão nos dados usados no treinamento. Em outras palavras é o momento em que tentamos obter a melhor configuração do "bias-variance tradeoff". No caso desse projeto, por existir desbalanceamento entre as classes a configuração de parâmetros foi escolhida com base no *f-measure* do modelo criado.

No caso das florestas aleatórias, é possível configurar uma quantidade considerável de parâmetros como: quantidade de árvores, profundidade máxima das árvores, etc....

O processo de configuração foi realizado usando *GridSearchCV*, onde os três parâmetros citados foram testados além do percentil de features a serem utilizadas. No final, apenas a quantidade de árvores (n_estimators) e a profundidade máxima das árvores (max_depth) foram mantidos. Mais especificamente:

- n_estimators = 10
- max_depth = 4
- percentile = 70%

Validação é a etapa em que o modelo é avaliado e seu desempenho é medido. É interessante que essa etapa seja realizada com um conjunto de dados diferente do que foi usado no treino de forma que o poder de generalização do modelo seja avaliado corretamente.

Nesse projeto, os dados foram divididos em conjuntos de treino e teste sendo 70% para treino e 30% para teste. O conjunto de treino passou por uma etapa de validação cruzada para a escolha da melhor configuração dos parâmetros. Por último, o melhor modelo foi avaliados nos dados de teste. É importante comentar que o processo de *oversampling* só foi aplicado ao conjunto de treino, dessa forma, o conjunto de teste só possui dados reais.

Como já foi comentado, o foco desse projeto foi melhorar as métricas *precision* e *recall*. A solução final obteve 31,5% de *precision* e 37.3% de *recall*. O primeiro resultado mostra que sempre que o modelo classificar alguém como POI, em 31,5% dos casos essa pessoa é realmente um POI. O segundo valor mostra que quando eu passar um POI como entrada para o modelo, existe uma chance de 37,3% de que esse POI seja classificado com um POI.