UDACITY

Return to "Data Analyst Nanodegree" in the classroom

Identify Fraud from Enron Email

REVIEW

CODE REVIEW 4

HISTORY

Meets Specifications

Prezado estudante,

Ótimo trabalho nesta submissão!
O relatório está bem claro e completo!
Minha maior sugestão tange a explicação do método de validação. Espero que a sugestão deixada seja útil!

Boa sorte no próximo projeto!

Qualidade do Código

O código reflete a descrição presente nas respostas das perguntas no relatório. O código faz as funções documentadas no relatório e o relatório especifica claramente a estratégia de análise final.

poi_id.py pode ser rodado para exportar o dataset, lista de features e algoritmo, de maneira que o algoritmo final pode ser verificado usando o tester.py.

Entendimento dos Dados e da Pergunta

A resposta do estudante trata as características mais importantes do conjunto de dados e usa estas características para fazer as suas análises.
Características importantes incluem:

- **número total de data points**
- **alocação entre classes (POI/non-POI)**
- **número de características usadas**
- **existem características com muitos valores faltando? etc.**

Dica: você pode utilizar visualizações para auxiliar nesse processo

BOM TRABALHO

Gostei da visualização!

O aluno identifica o(s) outlier(s) nos dados financeiros e explica como eles foram removidos ou tratados

BOM TRABALHO

Bom trabalho ao identificar os principais outliers que temos no dataset!

Otimização da Seleção de Características/Engenharia

Pelo menos uma feature foi implementada. A justificativa para ela foi dada nas respostas escritas e o efeito desta característica na performance final foi testada. O aluno não precisa incluir a nova característica no conjunto de dados final.

Bom trabalho ao criar novas features e apresentar os resultados obtidos com e sem elas!

Seleção de características univariadas ou recursivas foi feita ou as características foram escolhidas manualmente (diferente combinações de características foram feitas e o desempenho foi documentado para cada uma delas). Método de seleção e características selecionadas são documentadas e o número selecionado foi justificado. Para um algoritmo que suporta a verificação da importância das variáveis (ex. decision tree) ou pontuação das características (ex. SelectKBest), estas estão documentadas também.

BOM TRABALHO
Gostei do empenho! Muitos experimentos foram executados com diferentes configurações e números de features `k` para encontrar os melhores resultados!

Se o algoritmo requerir características com ajuste de escala, esta foi feita nos dados.

Escolha e Ajustes de um Algoritmo

Pelo menos 2 algoritmos diferentes são usados e seus desempenhos são comparados, com o de melhor desempenho sendo usado no modelo final. Essa comparação deve ser devidamente reportada no relatório

A resposta endereça o que significa fazer o afinamento(tuning) dos parâmetros e porque é importante fazê-lo.

Pelo menos um parâmetro importante com 3 valores é investigado sistematicamente, ou qualquer dos seguintes são verdadeiros:

- GridSearchCV usado para a busca do melhor parâmetro
- Vários parâmetros são afinados
- Busca de parâmetros incorporado na seleção do algoritmo (ex. parâmetros afinados para mais de um algoritmo e a melhor combinação algoritmo-parâmetros selecionada para a análise final).

Validar e Avaliar

Pelo menos duas métricas apropriadas são usadas para avaliar a performance do algoritmo (ex: precisão e abrangência - precision and recall), e o aluno explica o que estas métricas medem no contexto desta tarefa.

A resposta explica o que é validação e porque ela é importante.

O desempenho do modelo final é medido dividindo a base de dados entre base de treinamento e teste ou através do uso de validação cruzada (cross validation), especificando o tipo de validação usado.

SUGESTÃO

Neste projeto, estamos lidando com um dataset pequeno e desbalanceado.
Como vimos nas aulas, trabalhar com datasets pequenos é difícil, e para tornar a validação mais robusta, nós

normalmente usamos validação cruzada `k-fold`.
Mas veja, neste projeto (e muitos que vamos trabalhar com no dia a dia), uma validação estratificada **k-fold** é mais adequada.
A causa é que alguns folds podem ter muito poucos (ou até nenhum!) caso da classe minoritária. Desta forma, a estratificação garante que os conjuntos de treino e teste tenham o mesmo percentual de casos de cada classe.
Desta forma, sugiro ver esta página e esta também para mais (em inglês) e dar mais detalhes no relatório pq esse método em específico foi utilizado.

Quando tester.py é usado para avaliar a performance, precision e recall são, os dois, ao menos 0.3.

DOWNLOAD PROJECT

4 CODE REVIEW COMMENTS ›

RETURN TO PATH