

Documentação do Projecto

1. Visão Geral

A fase de refinamento do modelo aplicamos a técnica de GridSearchCV em conjunto da validação cruzada para capturar os melhores hiperparâmetros dos modelo em diferentes conjuntos de dados escolhidos aleatoriamente pelo `cross_val_score` e selecionamos algumas features para testar o desempenho do modelo.

2. Avaliação do Modelo

O nosso modelo obteve uma precisão de 76%, uma revogação de 67% uma acurácia de 80%, `f1_score` de 71% e na validação cruzada uma média de 65%. As áreas que precisam de melhoria são os resultados da precisão e revogação, pois essas são as métricas de negócio, embora a precisão e a revogação são inversamente proporcional isso quer dizer que se aumento uma reduz a outra ainda assim podemos melhorar os seus resultados, melhorando essas duas métricas por consequência conseguimos melhorar também o `f1_score` que é o equilíbrio entre as duas métricas.

3. Técnicas de refinamento

Como já descrito acima aplicamos a técnica de GridSearchCV para o ajuste de hiperparâmetros, seleção de features com a técnica de seleção Passo a Passo e Filtro Estatístico Bivariado, testamos os algoritmos baseados em vizinhos mais próximos(KNN), árvores de decisão(XGBoost) e o algoritmo base de regressão logística, testamos diferentes agrupamentos de dados com a técnica de validação cruzada(`cross_val_score`) com `cv = 3` sendo esse numero de agrupamento que obteve o melhor desempenho para os modelos em geral.

4. Ajuste de hiperparâmetros

Para a nossa solução não tivemos a necessidade de realizar qualquer ajuste adicional dos hiperparâmetros.

5. Validação cruzada

Para a validação cruzada testamos alguns números de grupos, como a nossa base de dados tem 119390 registros optamos por 5 grupos, depois 6 mas os modelos tinham a tendência de obter uma média da validação cruzada reduzida a medida que os números de grupos subiam, então começamos a reduzir o número de agrupamento e o inverso começou a acontecer os nossos modelos obtiveram uma média alta com 3 grupos de conjunto de dados(`cv=3`) porque assim para cada grupo de dados tinham registros suficientes para os modelos extrair padrões e obter bons resultados.

6. Seleção de Recursos

Para a seleção de features aplicamos a técnica de seleção passo a passo que selecionou as melhores features para o modelo, mas no nosso conjunto de dados teve uma coluna “estado_reserva” era uma coluna que é basicamente uma cópia da nossa coluna alvo, e essa técnica deixou escapar porque contribuiu bastante nas previsões e isso fazia com que os modelos tivessem overfitting tanto nos dados de teste e validação, após alguns testes de correlação eliminamos essa coluna manualmente, aplicamos também a seleção baseada em Filtro Estatístico Bivariado que aplicamos as colunas categóricas porque elas também foram crucial para adicionar mais informações aos modelos, antes de realizar qualquer transformação de Encoder as colunas categóricas decidimos realizar primeiro essa seleção para garantir que estamos a aplicar essa transformação somente em colunas categóricas relevantes. Essas técnicas são relevantes para o modelo porque sem informações que estão correlacionadas a variável alvo, o modelo aprende padrões irrelevantes para as previsões e o modelo obtém um desempenho ruim tanto na fase do treinamento quanto na fase de previsão.

Envio de Teste

1. Visão geral

A fase de envio do teste é o momento em que os dados são tratados para que possam estar nos mesmos padrões dos dados de treinamento do modelo, com o conhecimento que o modelo adquiriu no conjunto de treinamento o modelo deverá fazer previsões mais próximas da realidade, tendo o mínimo de erro possível.

2. Preparação de dados para teste

Para a fase de teste é necessário garantir que os dados de predição não contenha valores nulos, deverá ser feito a seleção de features e a transformação de colunas categóricas.

3. Aplicação do modelo

Após a preparação dos dados de teste o modelo fez as previsões com o método `predict(knn.predict)`:

```
knn_predict = knn.predict(Xtest)
```

4. Métricas de teste

As métricas de avaliação foram a precisão com 73%, revogação com 67% e o f1_score com 70%, em comparação aos dados de treinamento e validação a precisão caiu um pouco com uma diferença de 3% a revogação se manteve igual e o f1_score teve uma diferença de 1%.

5. Implantação de código

O modelo pode ser levado a produção com um container e ser integrado com uma plataforma de gestão de um hotel e realizar as previsões.

6. Implementação de código

```
from sklearn.model_selection import GridSearchCV, StratifiedKFold
from sklearn.metrics import precision_score, make_scorer
```

```
# 1. Instanciar o modelo base
knn = KNeighborsClassifier()

# 2. Definir a grade de parâmetros para KNeighborsClassifier
param_grid_knn = {
    'n_neighbors': [3, 5, 7, 9, 11], # Testar um conjunto de vizinhos ímpares
    'weights': ['uniform', 'distance'] # Dois métodos comuns
}
```

```
# 3. Instanciar o GridSearchCV (usando seu scorer e kf já definidos)
grid_search_knn = GridSearchCV(
    estimator=knn,
    param_grid=param_grid_knn,
    scoring=scorer,
    cv=kf,
    verbose=1,
    n_jobs=-1
)

# 4. Executar a busca
grid_search_knn.fit(X, y)
```

```
# 5. Resultados
print("\n--- KNeighbors Otimizado ---")
print(f"Melhor Score de Recall (CV): {grid_search_knn.best_score_:.4f}")
print("Melhores Parâmetros Encontrados:")
print(grid_search_knn.best_params_)
best_knn = grid_search_knn.best_estimator_

Fitting 5 folds for each of 10 candidates, totalling 50 fits

--- KNeighbors Otimizado ---
Melhor Score de Recall (CV): 0.7689
Melhores Parâmetros Encontrados:
{'n_neighbors': 11, 'weights': 'distance'}
```

Conclusão

A fase de refinamento é o momento que demos mais atenção as parâmetros do modelo para ter o melhor resultado possível com os dados de teste, o maior desafio encontrado foi poder identificar o vies nos dados que estava a provocar o overfitting no modelo e quais features estavam a contribuir significativamente para a predição de qualidade do modelo, o desempenho ainda não é o melhor possível ainda podemos tratar ainda mais os dados melhorando a qualidade, pois acreditamos que com um dado de qualidade até o menor modelo pode alcançar resultados muito extraordinário.

Referências

1. <https://matheus-buniotto.medium.com/usando-automl-para-previs%C3%A3o-de-demanda-e-cancelamento-de-hospedagem-hotel-e-resort-f2ac289ec7ac>
2. <https://www.mdpi.com/2071-1050/17/5/2210>
3. <https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/discussion/565098>