

Aprendizado de Máquina (Algoritmos de ML) - Otimização e Validação do Modelo

1. Otimização dos Hiperparâmetros dos Modelos para Melhorar a Performance

Utilize Técnicas como Grid Search para Encontrar os Melhores Hiperparâmetros.

A otimização dos hiperparâmetros é essencial para melhorar a performance dos modelos de aprendizado de máquina. Grid Search é uma técnica comum usada para este propósito, onde um conjunto de hiperparâmetros é testado de forma exaustiva para encontrar a combinação que proporciona o melhor desempenho.

Exemplo de Otimização com Grid Search:

```
from sklearn.model_selection import GridSearchCV

# Definir os parâmetros a serem testados
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth': [10, 20, 30, None]
}

# Configurar o Grid Search
grid_search = GridSearchCV(estimator=RandomForestClassifier(random_state=42),
    param_grid=param_grid, cv=5, scoring='accuracy')

# Realizar o Grid Search
grid_search.fit(X_train, y_train)

# Obter os melhores parâmetros
best_params = grid_search.best_params_
print(f"Melhores parâmetros: {best_params}")

# Treinar o modelo com os melhores parâmetros
best_model = RandomForestClassifier(**best_params, random_state=42)
best_model.fit(X_train, y_train)
```

2. Validação Cruzada para Verificar a Robustez do Modelo

Realize Validação Cruzada para Avaliar o Desempenho em Diferentes Conjuntos de Dados. A validação cruzada é uma técnica usada para avaliar a robustez de um modelo, dividindo o conjunto de dados em várias partes e treinando e testando o modelo em diferentes subconjuntos.

Exemplo de Validação Cruzada:

```
from sklearn.model_selection import cross_val_score

# Avaliar o modelo com validação cruzada
cv_scores = cross_val_score(best_model, X, y, cv=5, scoring='accuracy')

# Exibir os resultados da validação cruzada
print(f"Scores da validação cruzada: {cv_scores}")
print(f"Média dos scores: {cv_scores.mean()}")
print(f"Desvio padrão dos scores: {cv_scores.std()}")
```

3. Documentação do Processo de Construção e Treinamento do Modelo

Projeto: HealthTech Balderi Solutions - Previsão de Diagnósticos Médicos

Objetivo: Desenvolver um modelo de aprendizado de máquina para prever diagnósticos médicos com base em dados de pacientes e exames.

Passos Realizados:

1. Coleta e Pré-processamento de Dados:

- 1.1 Coleta de dados de pacientes, consultas e exames.
- 1.2 Limpeza e pré-processamento, incluindo tratamento de valores ausentes e outliers, e padronização de formatos.

2. Escolha de Algoritmo:

2.1 Random Forest foi escolhido devido à sua robustez e capacidade de lidar com dados de alta dimensionalidade.

3. Implementação do Modelo:

3.1 Modelo treinado utilizando dados de glicose, pressão sanguínea e IMC.

4. Otimização dos Hiperparâmetros:

4.1 Grid Search foi utilizado para encontrar os melhores hiperparâmetros.

4.2 Melhores parâmetros encontrados: n_estimators=200, max_features='auto', max_depth=20.

5. Validação Cruzada:

5.1 Validação cruzada com 5 folds foi realizada para avaliar a robustez do modelo.

5.2 Scores da validação cruzada: [0.92, 0.90, 0.91, 0.93, 0.89]

5.3 Média dos scores: 0.91 • Desvio padrão dos scores: 0.015

Resultados:

Random Forest Classifier com `n_estimators=200`, `max_features='auto'`, `max_depth=20`.

Accuracy: 92%

Precision: 91% •

Recall: 90%

F1-Score: 90.5