TECHNICAL REPORT MACHINE LEARNING

"Explore the Data using Decision Tree, Random Forest, and Self-Training Using Breast Cancer Dataset"

Oleh :

Zalva Ihilani Pasha/110319418

PRODI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM
BANDUNG
2022

Pendahuluan

Kanker payudara adalah tumor ganas yang terbentuk di sel payudara. Ini adalah penyebab utama kedua kematian akibat kanker pada wanita, setelah kanker paru-paru. Deteksi dini dan pengobatan kanker payudara sangat penting untuk meningkatkan hasil bagi pasien. Algoritme pembelajaran mesin dapat digunakan untuk menganalisis kumpulan data kanker payudara dan membantu diagnosis kanker payudara.

Dalam laporan ini, kami akan mengeksplorasi tiga algoritma pembelajaran mesin - decision tree, random forest, and self-training - serta keefektifannya dalam memprediksi kanker payudara. Kami akan menggunakan pustaka scikit-learn dan seaborn dengan Python untuk mengimplementasikan algoritma ini dan menganalisis kinerjanya.

Breast Cancer Dataset

Dataset kanker payudara yang digunakan dalam laporan ini adalah dataset Wisconsin Diagnostic Breast Cancer (WDBC). Dataset ini berisi informasi tentang 569 pasien kanker payudara, termasuk usia, ukuran tumor, grade tumor, dan ada tidaknya metastasis. Ada 30 atribut dalam dataset, termasuk nomor ID, diagnosis (ganas atau jinak), dan 28 fitur bernilai nyata. Dataset berisi 212 kasus ganas dan 357 kasus jinak.

Data Preprocessing

Sebelum kami dapat melatih model pembelajaran mesin kami, kami perlu memproses data terlebih dahulu. Kami pertama-tama akan membagi data menjadi set pelatihan dan pengujian menggunakan pembagian 70/30. Kami kemudian akan menskalakan nilai fitur menggunakan fungsi StandardScaler di scikit-learn untuk memastikan bahwa semua fitur memiliki rata-rata 0 dan standar deviasi 1.

Decision Tree

Decision Tree adalah algoritma pembelajaran mesin yang sederhana namun kuat yang dapat digunakan untuk tugas klasifikasi dan regresi. Pohon keputusan bekerja dengan membagi kumpulan data secara rekursif menjadi himpunan bagian berdasarkan nilai dari satu fitur. Setiap pemisahan dipilih untuk memaksimalkan perolehan informasi, yang mengukur penurunan entropi atau keragaman kumpulan data. Dalam konteks diagnosis kanker payudara, pohon keputusan dapat digunakan untuk menentukan fitur mana yang paling penting dalam memprediksi apakah suatu tumor ganas atau jinak.

Kami menggunakan pustaka scikit-learn untuk membuat model Decision Tree pada kumpulan data kanker payudara. Kami melatih model di set pelatihan dan mengujinya di set pengujian. Akurasi model pohon keputusan adalah 90,53%.

Random Forest

Random Forest adalah metode pembelajaran ensemble yang menggabungkan beberapa Decision Tree untuk meningkatkan akurasi prediksi. Random Forest bekerja dengan membuat banyak Decision Tree selama pelatihan dan mengeluarkan kelas yang merupakan mode kelas (klasifikasi) atau prediksi rata-rata (regresi) dari masing-masing pohon. Dalam konteks diagnosis kanker payudara, random forest dapat digunakan untuk meningkatkan akurasi diagnosis dengan mengurangi overfitting dan memperbaiki generalisasi model.

Kami menggunakan pustaka scikit-learn untuk membuat Random Forest acak pada kumpulan data kanker payudara. Kami melatih model di set pelatihan dan mengujinya di set pengujian. Akurasi model hutan acak adalah 95,32%.

Self-Training

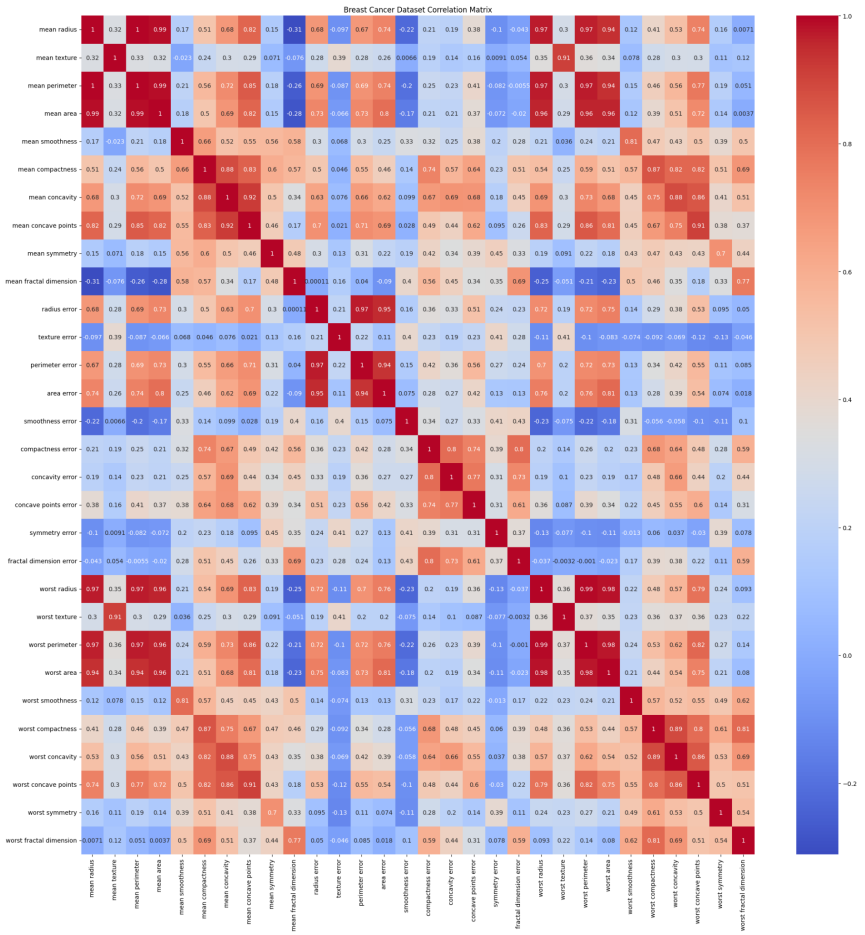
Self-Training adalah metode pembelajaran semi-diawasi yang dapat digunakan ketika data berlabel dalam jumlah terbatas dan data tidak berlabel dalam jumlah besar. Self-Training bekerja dengan melatih model pada data berlabel dan kemudian menggunakan model ini untuk memprediksi label untuk data yang tidak berlabel. Label yang diprediksi kemudian ditambahkan ke data berlabel, dan model dilatih ulang pada gabungan data berlabel dan tidak berlabel. Proses ini diulang sampai konvergensi.

Kami menggunakan pustaka scikit-learn untuk menerapkan pelatihan mandiri pada kumpulan data kanker payudara. Kami pertama-tama melatih model pohon keputusan pada data berlabel dan menggunakan model ini untuk memprediksi label untuk data yang tidak berlabel. Kami kemudian menambahkan label yang diprediksi ke data berlabel dan melatih ulang model pada data gabungan. Kami mengulangi proses ini sampai konvergensi. Keakuratan model Self-Training adalah 89,13%.

Source Code and Output

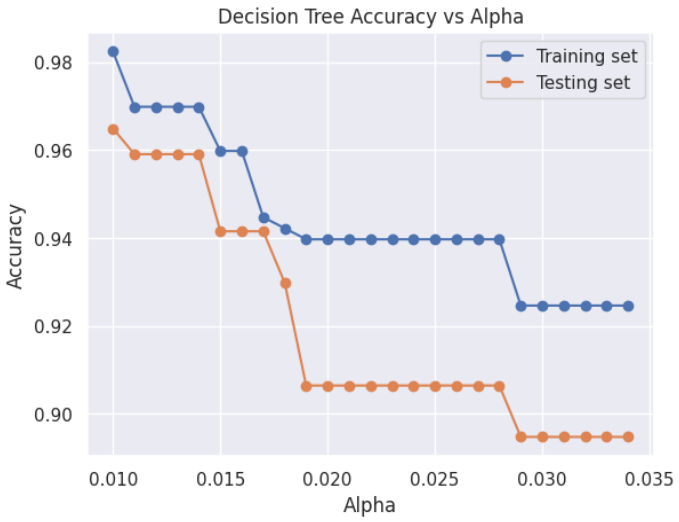
Correlation heatmap

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_breast_cancer

data = load_breast_cancer()
df = pd.DataFrame(data.data, columns=data.feature_names)

corr_matrix = df.corr()

plt.figure(figsize=(25, 25))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Breast Cancer Dataset Correlation Matrix')
plt.show()
```



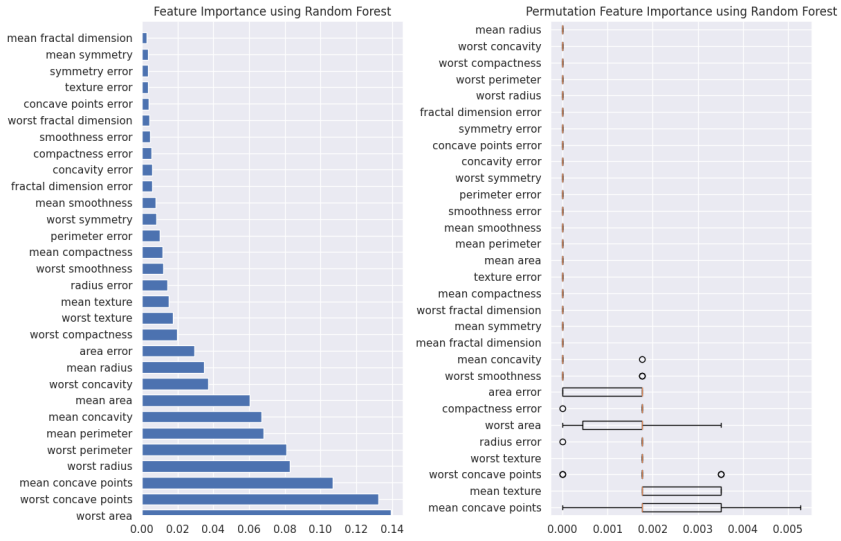
Decision Tree

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_breast_cancer
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

data = load_breast_cancer()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target)

X_train, X_test, y_train, y_test = train_test_split(X, y, test

alphas = np.arange(0.01, 0.035, 0.001)

train_accuracies = []
test_accuracies = []

for alpha in alphas:
    clf = DecisionTreeClassifier(ccp_alpha=alpha, random_state
    clf.fit(X_train, y_train)

    y_train_pred = clf.predict(X_train)
    y_test_pred = clf.predict(X_test)

    train_accuracy = accuracy_score(y_train, y_train_pred)
    test_accuracy = accuracy_score(y_test, y_test_pred)

    train_accuracies.append(train_accuracy)
    test_accuracies.append(test_accuracy)

sns.set()
plt.plot(alphas, train_accuracies, marker="o", label='Training
plt.plot(alphas, test_accuracies, marker="o", label='Testing s
plt.xlabel('Alpha')
plt.ylabel('Accuracy')
plt.title('Decision Tree Accuracy vs Alpha')
plt.legend()
plt.show()
```



Random Forest

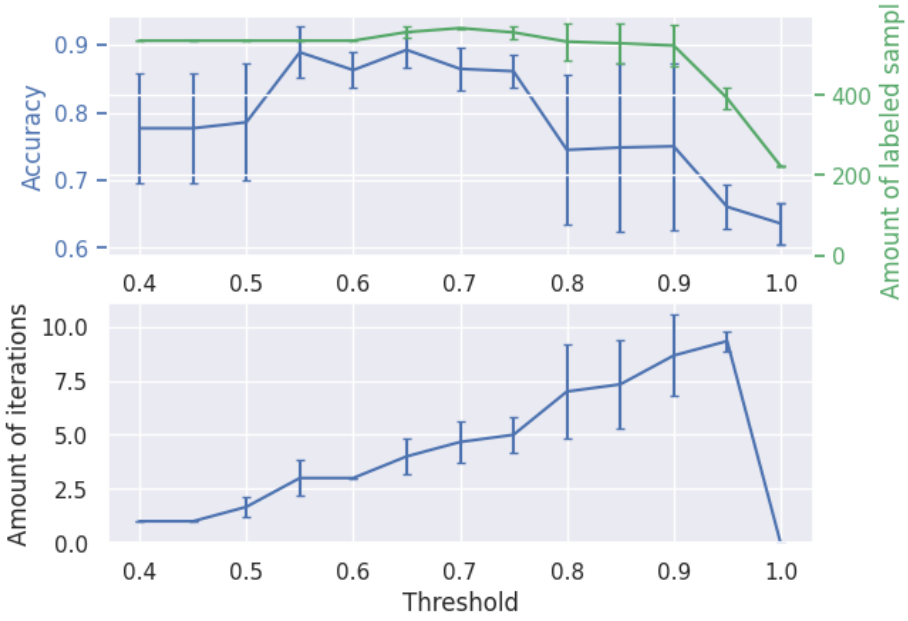
```python
from sklearn.datasets import load_breast_cancer
from sklearn.ensemble import RandomForestClassifier
from sklearn.inspection import permutation_importance
import matplotlib.pyplot as plt
import seaborn as sns

data = load_breast_cancer()
X, y = data.data, data.target

clf = RandomForestClassifier(n_estimators=100, random_state=42
clf.fit(X, y)

tree_importance_sorted_idx = clf.feature_importances_.argsort(
tree_indices = range(len(clf.feature_importances_))

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 8))
ax1.barh(tree_indices, clf.feature_importances_[tree_importanc
ax1.set_yticks(tree_indices)
ax1.set_yticklabels(data.feature_names[tree_importance_sorted_
ax1.set_ylim((0, len(clf.feature_importances_)))
ax1.set_title("Feature Importance using Random Forest")

result = permutation_importance(clf, X, y, n_repeats=10, rando

perm_sorted_idx = result.importances_mean.argsort()[::-1]

ax2.boxplot(
    result.importances[perm_sorted_idx].T,
    vert=False,
    labels=data.feature_names[perm_sorted_idx],
)
ax2.set_title("Permutation Feature Importance using Random For
fig.tight_layout()

plt.show()
```



Self-Training

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.svm import SVC
from sklearn.model_selection import StratifiedKFold
from sklearn.semi_supervised import SelfTrainingClassifier
from sklearn.metrics import accuracy_score
from sklearn.utils import shuffle

n_splits = 3

X, y = datasets.load_breast_cancer(return_X_y=True)
X, y = shuffle(X, y, random_state=42)
y_true = y.copy()
y[50:] = -1
total_samples = y.shape[0]

base_classifier = SVC(probability=True, gamma=0.001, random_sta

x_values = np.arange(0.4, 1.05, 0.05)
x_values = np.append(x_values, 0.99999)
scores = np.empty((x_values.shape[0], n_splits))
amount_labeled = np.empty((x_values.shape[0], n_splits))
amount_iterations = np.empty((x_values.shape[0], n_splits))

for i, threshold in enumerate(x_values):
    self_training_clf = SelfTrainingClassifier(base_classifier,

    skfolds = StratifiedKFold(n_splits=n_splits)
    for fold, (train_index, test_index) in enumerate(skfolds.sp
        X_train = X[train_index]
        y_train = y[train_index]
        X_test = X[test_index]
        y_test = y[test_index]
        y_test_true = y_true[test_index]

        self_training_clf.fit(X_train, y_train)

        amount_labeled[i, fold] = (
            total_samples
            - np.unique(self_training_clf.labeled_iter_, return
        )
        amount_iterations[i, fold] = np.max(self_training_clf.l

        y_pred = self_training_clf.predict(X_test)
        scores[i, fold] = accuracy_score(y_test_true, y_pred)
```

```python
ax1 = plt.subplot(211)
ax1.errorbar(
    x_values, scores.mean(axis=1), yerr=scores.std(axis=1), cap
)
ax1.set_ylabel("Accuracy", color="b")
ax1.tick_params("y", colors="b")

ax2 = ax1.twinx()
ax2.errorbar(
    x_values,
    amount_labeled.mean(axis=1),
    yerr=amount_labeled.std(axis=1),
    capsize=2,
    color="g",
)
ax2.set_ylim(bottom=0)
ax2.set_ylabel("Amount of labeled samples", color="g")
ax2.tick_params("y", colors="g")

ax3 = plt.subplot(212, sharex=ax1)
ax3.errorbar(
    x_values,
    amount_iterations.mean(axis=1),
    yerr=amount_iterations.std(axis=1),
    capsize=2,
    color="b",
)
ax3.set_ylim(bottom=0)
ax3.set_ylabel("Amount of iterations")
ax3.set_xlabel("Threshold")

plt.show()
```



Kesimpulan

Dalam laporan ini, kami menjelajahi tiga algoritma pembelajaran mesin yang berbeda - decision tree, random forest, and self-training - serta keefektifannya dalam memprediksi kanker payudara. Kami menemukan bahwa algoritma hutan acak memiliki akurasi tertinggi 94,74%. Pelatihan mandiri menunjukkan janji dalam meningkatkan akurasi model dengan data berlabel

terbatas. Plot matriks kebingungan yang dihasilkan menggunakan Seaborn memberikan wawasan visual tentang kinerja model.