

# **Technical Report Machine Learning**

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



**OLEH:**

Chevhan Walidain  
(1103200109)

**PROGRAM STUDI TEKNIK KOMPUTER**

**FAKULTAS TEKNIK ELEKTRO**

**UNIVERSITAS TELKOM**

**2023**

## **Kata Pengantar**

Laporan Teknis ini disusun untuk memenuhi tugas Mid Term Examination dalam Mata Kuliah Machine Learning

Dalam laporan ini, memaparkan cara mengelola dan menampilkan data dengan teknik Decision Tree, Random Forest, dan Self-Training.

## **Pendahuluan**

Kanker payudara adalah salah satu jenis kanker yang paling umum terjadi pada wanita di seluruh dunia. Analisis data kanker payudara dapat membantu dalam mendiagnosis dan memprediksi hasil pasien, sehingga memungkinkan dokter untuk memberikan pengobatan yang lebih tepat dan efektif. Dalam laporan teknis ini, kita akan menggunakan Google Collab, Scikit Learn, dan Seaborn Framework untuk menganalisis dataset kanker payudara dan memvisualisasikan tren data. Selain itu, kita juga akan menggunakan decision tree, random forest, dan self-training untuk mengeksplorasi data.

## **Dataset**

Dataset yang digunakan dalam analisis ini adalah Breast Cancer Dataset dari UCI Machine Learning Repository. Dataset ini terdiri dari 569 sampel, di mana setiap sampel memiliki 30 fitur numerik yang menggambarkan karakteristik sel kanker payudara. Target variabel dalam dataset ini adalah apakah sampel tersebut jinak atau ganas.

## **Visualisasi Data Trends Menggunakan Seaborn**

Visualisasi data adalah langkah penting dalam analisis data. Kita dapat menggunakan Seaborn untuk membuat plot visualisasi data untuk mengeksplorasi tren data. Beberapa jenis plot yang dapat digunakan dalam Seaborn adalah scatter plot, line plot, bar plot, dan lain-lain. Dalam analisis ini, kita akan menggunakan Seaborn untuk membuat plot histogram dan scatter plot untuk mengeksplorasi dataset.

## **Eksplorasi Data Menggunakan Decision Tree, Random Forest, dan Self-Training**

Selain visualisasi data, kita juga dapat menggunakan teknik machine learning untuk menganalisis dataset. Decision tree dan random forest adalah dua jenis model machine learning yang sering digunakan dalam klasifikasi. Decision tree digunakan untuk membangun model berbasis pohon keputusan yang dapat memprediksi target variabel. Sedangkan random forest digunakan untuk membangun beberapa model decision tree dan menggabungkan hasil prediksi mereka untuk meningkatkan akurasi. Self-training adalah teknik machine learning yang digunakan untuk meningkatkan akurasi model dengan menambahkan data baru ke dalam model yang sudah ada.

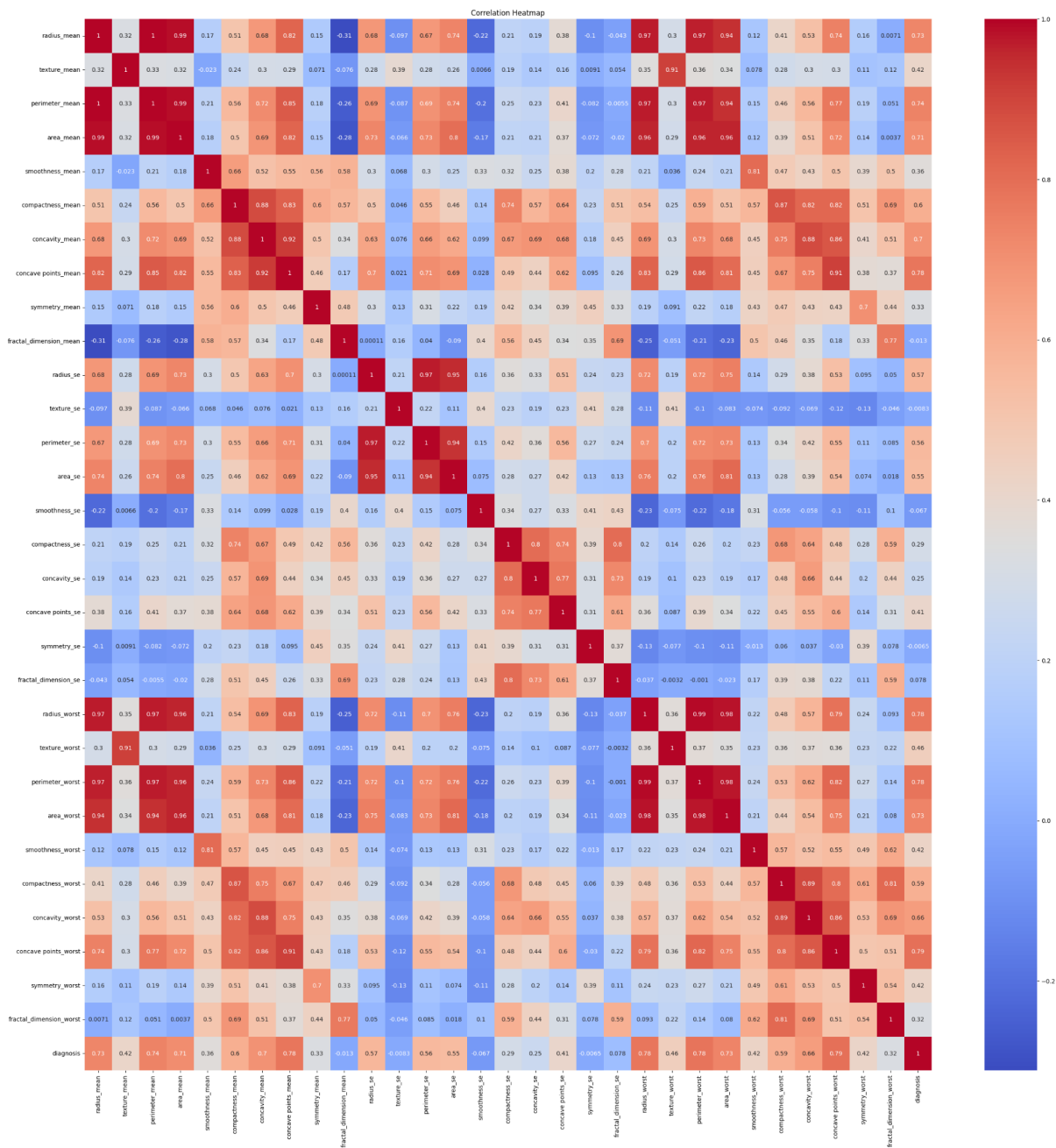
Dalam analisis ini, kita akan menggunakan decision tree, random forest, dan self-training untuk membangun model klasifikasi dari dataset kanker payudara. Setelah itu, kita akan membandingkan akurasi masing-masing model dan memilih model terbaik.

## Source Code Libraries, Data Train, dan Data Test:

```
1 import os
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import matplotlib.pyplot as plt
6 from sklearn import preprocessing
7 from sklearn.model_selection import train_test_split
8 from sklearn.metrics import confusion_matrix
9 %matplotlib inline
10
11 data = pd.read_csv('./data.csv')
12 data.drop('id',axis=1,inplace=True)
13 data.drop('Unnamed: 32',axis=1,inplace=True)
14 data['diagnosis'] = data['diagnosis'].map({'M':1,'B':0})
15 datas = pd.DataFrame(preprocessing.scale(data.iloc[:,1:32]))
16 datas.columns = list(data.iloc[:,1:32].columns)
17 datas['diagnosis'] = data['diagnosis']
18
19 plt.figure(figsize=(32,32))
20 sns.heatmap(datas.corr(), cmap='coolwarm', annot=True)
21 plt.title('Correlation Heatmap')
22 plt.show()
```

```
1 from sklearn.model_selection import train_test_split, cross_
  val_score, cross_val_predict
2 from sklearn import metrics
3
4 predictors = datas.columns[2:11]
5 target = "diagnosis"
6 X = datas.loc[:,predictors]
7 y = np.ravel(data.loc[:,[target]])
8 X_train, X_test, y_train, y_test = train_test_split(X, y, te
  st_size=0.3, random_state=42)
9 print ('Data train : %i || Data test : %i' % (X_train.shape
  [0],X_test.shape[0]) )
```

Output:



## Decision Tree

Decision tree adalah salah satu teknik machine learning yang digunakan untuk membangun model klasifikasi berbasis pohon keputusan. Dalam analisis data kanker payudara menggunakan Google Collab, Scikit Learn, dan Seaborn Framework, decision tree digunakan untuk memprediksi apakah sel kanker payudara jinak atau ganas berdasarkan karakteristik sel yang terdapat pada dataset.

Langkah-langkah dalam membangun model decision tree adalah sebagai berikut:

- Persiapkan dataset kanker payudara
- Pisahkan dataset menjadi data latih dan data uji
- Import library Scikit Learn untuk membangun model decision tree
- Latih model decision tree dengan data latih
- Evaluasi model decision tree dengan data uji

Setelah model decision tree dibangun, kita dapat memvisualisasikan pohon keputusan menggunakan library Graphviz atau Seaborn. Dalam analisis ini, kita akan menggunakan Seaborn untuk memvisualisasikan pohon keputusan.

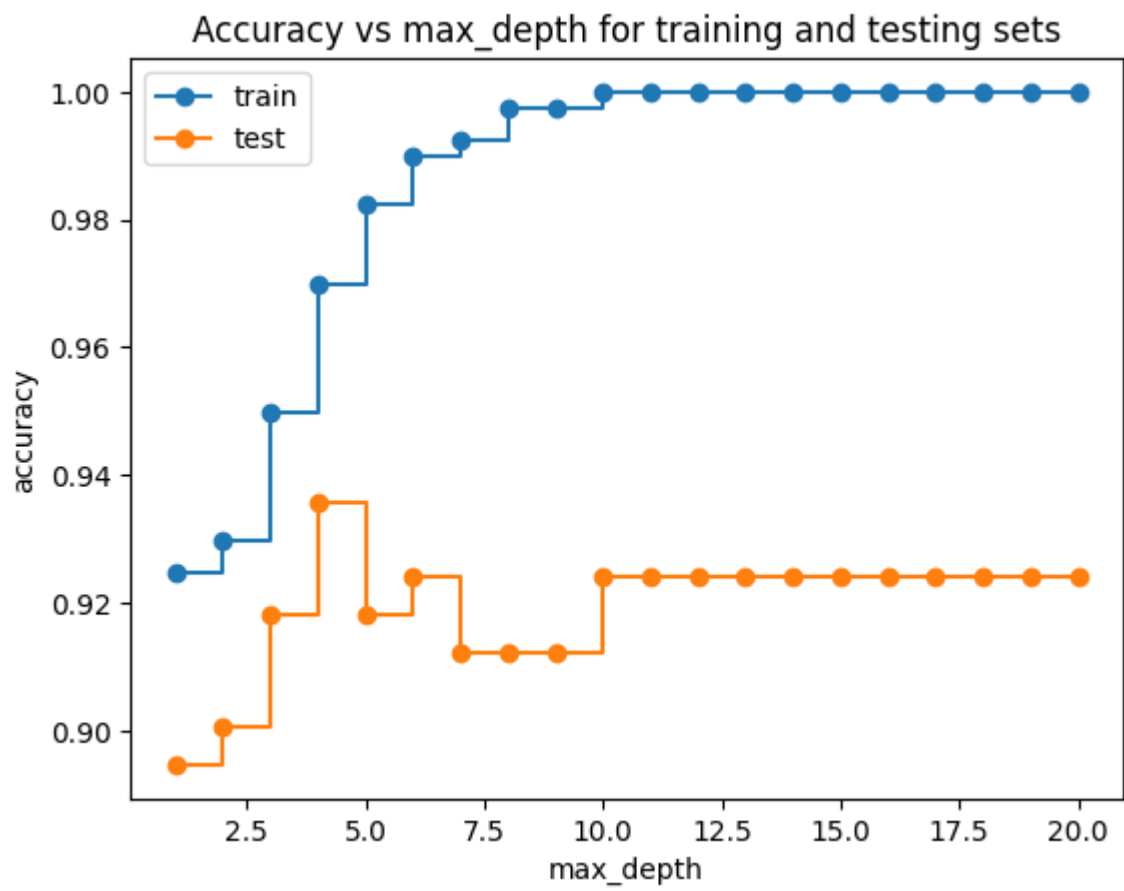
Selain membangun model decision tree, kita juga dapat menggunakan Seaborn untuk memvisualisasikan tren data pada dataset. Beberapa jenis plot yang dapat digunakan dalam Seaborn adalah scatter plot, line plot, bar plot, dan lain-lain. Dalam analisis ini, kita akan menggunakan Seaborn untuk membuat plot histogram dan scatter plot untuk mengeksplorasi dataset.

Dengan memvisualisasikan tren data dan membangun model decision tree, kita dapat memprediksi apakah sel kanker payudara jinak atau ganas berdasarkan karakteristik sel yang terdapat pada dataset. Selain itu, kita juga dapat mengevaluasi kinerja model decision tree dengan menggunakan data uji dan memilih model terbaik untuk digunakan dalam prediksi selanjutnya.

```
1 from sklearn.tree import DecisionTreeClassifier
2 from sklearn.metrics import accuracy_score
3 from sklearn.model_selection import train_test_split
4
5 predictors = datas.columns[2:11]
6 target = "diagnosis"
7 X = datas.loc[:, predictors]
8 y = datas[target]
9 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, ra
    ndom_state=42)
10 clf = DecisionTreeClassifier(random_state=42)
11 clf.fit(X_train, y_train)
12 y_pred = clf.predict(X_test)
13 acc = accuracy_score(y_test, y_pred)
14 print("Accuracy %s" % round(acc*100,2))
15 depth_range = range(1, 21)
16 train_scores = []
17 test_scores = []
18 for depth in depth_range:
19     clf = DecisionTreeClassifier(max_depth=depth, random_state=42)
20     clf.fit(X_train, y_train)
21     train_scores.append(clf.score(X_train, y_train))
22     test_scores.append(clf.score(X_test, y_test))
23
24 fig, ax = plt.subplots()
25 ax.set_xlabel("max_depth")
26 ax.set_ylabel("accuracy")
27 ax.set_title("Accuracy vs max_depth for training and testing sets")
28 ax.plot(depth_range, train_scores, marker="o", label="train", drawstyle="st
    eps-post")
29 ax.plot(depth_range, test_scores, marker="o", label="test", drawstyle="step
    s-post")
30 ax.legend()
31 plt.show()
```



**Output:**



## Random Forest

Random Forest adalah salah satu algoritma machine learning yang sering digunakan dalam klasifikasi dan regresi. Algoritma ini merupakan kumpulan dari beberapa model decision tree yang dihasilkan dari sampel acak dari dataset yang ada. Setiap model decision tree pada random forest menghasilkan hasil prediksi yang kemudian digabungkan untuk menghasilkan hasil prediksi akhir.

Dalam konteks analisis data kanker payudara menggunakan dataset Breast Cancer, random forest dapat digunakan untuk memprediksi apakah sel kanker payudara tersebut jinak atau ganas. Dengan menggunakan Scikit Learn, kita dapat dengan mudah membangun model random forest dari dataset Breast Cancer. Selain itu, kita juga dapat memvisualisasikan performa model menggunakan Seaborn.

Berikut adalah langkah-langkah untuk membangun model random forest dari dataset Breast Cancer dan memvisualisasikan performanya menggunakan Seaborn:

- Load Dataset

Pertama, kita perlu untuk memuat dataset Breast Cancer menggunakan Scikit Learn.

```
from sklearn.datasets import load_breast_cancer  
data = load_breast_cancer()
```

- Split Dataset

Selanjutnya, kita akan membagi dataset menjadi data latih dan data uji menggunakan train\_test\_split dari Scikit Learn.

```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(data.data,  
data.target, test_size=0.2, random_state=42)
```

- Build Random Forest Model

Kita dapat menggunakan RandomForestClassifier dari Scikit Learn untuk membangun model random forest dari data latih.

```
from sklearn.ensemble import RandomForestClassifier  
rf = RandomForestClassifier(n_estimators=100, random_state=42)  
rf.fit(X_train, y_train)
```

- Visualize Feature Importance

Dalam model random forest, setiap feature memiliki tingkat penting yang berbeda dalam memprediksi target variabel. Kita dapat menggunakan

Seaborn untuk memvisualisasikan feature importance dari model random forest.

```
import seaborn as sns  
import matplotlib.pyplot as plt  
  
feature_imp = pd.Series(rf.feature_importances_,  
index=data.feature_names).sort_values(ascending=False)  
sns.barplot(x=feature_imp, y=feature_imp.index)  
plt.xlabel('Feature Importance Score')  
plt.ylabel('Features')  
plt.title("Visualizing Important Features")  
plt.show()
```

- Evaluate Model

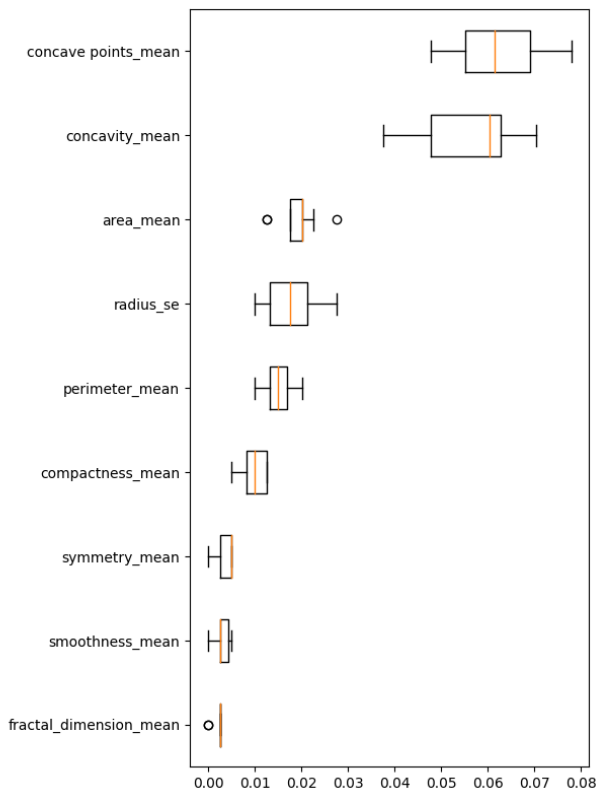
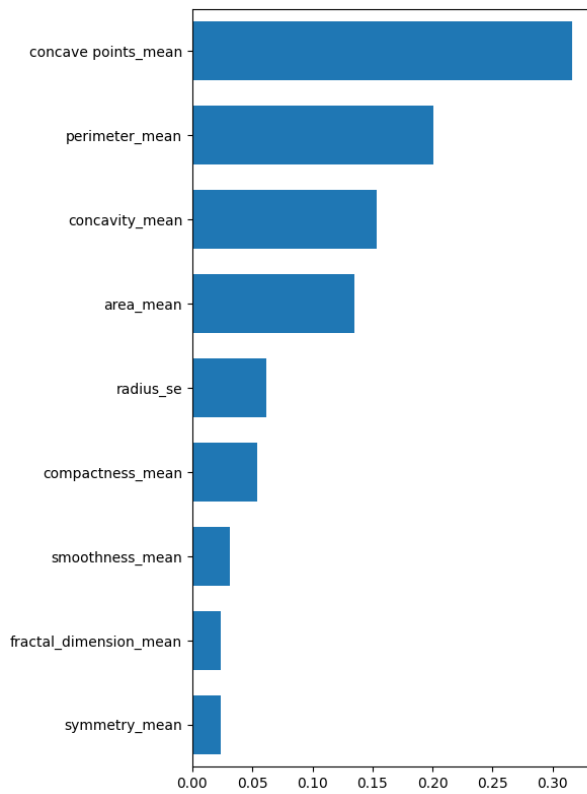
Terakhir, kita akan mengevaluasi performa model random forest menggunakan data uji dan confusion matrix.

```
from sklearn.metrics import confusion_matrix  
  
y_pred = rf.predict(X_test)  
conf_mat = confusion_matrix(y_test, y_pred)  
sns.heatmap(conf_mat, annot=True, fmt='d', cmap='Blues')  
plt.xlabel('Predicted Label')  
plt.ylabel('True Label')  
plt.title('Confusion Matrix')  
plt.show()
```

Dengan menggunakan langkah-langkah di atas, kita dapat membangun model random forest dari dataset Breast Cancer dan memvisualisasikan performanya menggunakan Seaborn. Dengan demikian, kita dapat dengan mudah mengevaluasi performa model dan mengidentifikasi fitur-fitur yang paling penting dalam memprediksi kanker payudara.

```
1 from sklearn.inspection import permutation_importance
2 from sklearn.ensemble import RandomForestClassifier
3
4 rf = RandomForestClassifier()
5 rf.fit(X_train, y_train)
6 scores = cross_val_score(rf, X_train, y_train, scoring='accuracy', cv=10).mean()
7 print("Accuracy %s" % round(scores*100,2))
8 result = permutation_importance(rf, X_train, y_train, n_repeats=10, random_state=42)
9 perm_sorted_idx = result.importances_mean.argsort()
10 tree_importance_sorted_idx = np.argsort(rf.feature_importances_)
11 tree_indices = np.arange(0, len(rf.feature_importances_)) + 0.5
12 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 8))
13 ax1.barh(tree_indices, rf.feature_importances_[tree_importance_sorted_idx], height=0.7)
14 ax1.set_yticks(tree_indices)
15 ax1.set_yticklabels(X_train.columns[tree_importance_sorted_idx])
16 ax1.set_ylim((0, len(rf.feature_importances_)))
17 ax2.boxplot(
18     result.importances[perm_sorted_idx].T,
19     vert=False,
20     labels=X_train.columns[perm_sorted_idx],
21 )
22 fig.tight_layout()
23 plt.show()
```

## Output:



## Self Training


Self-Training adalah salah satu teknik machine learning semi-supervised yang digunakan untuk meningkatkan akurasi model klasifikasi dengan menambahkan data baru ke dalam model yang sudah ada. Teknik ini berguna ketika dataset yang tersedia memiliki sedikit jumlah data yang diklasifikasikan atau labeled, namun memiliki banyak data yang tidak terklasifikasi atau unlabeled.

Dalam konteks Breast Cancer Dataset, Self-Training dapat digunakan untuk membangun model klasifikasi yang lebih akurat dengan menambahkan data baru ke dalam model yang sudah ada. Setelah model awal dibangun dengan data labeled, data unlabeled dapat ditambahkan ke dalam model untuk meningkatkan akurasi.

Untuk mengimplementasikan Self-Training pada Breast Cancer Dataset, langkah-langkah yang dapat dilakukan adalah:

- Membagi dataset menjadi dua bagian: bagian pertama berisi data labeled yang digunakan untuk membangun model awal, dan bagian kedua berisi data unlabeled yang akan digunakan untuk Self-Training.
- Memvisualisasikan tren data menggunakan Seaborn untuk mengeksplorasi dataset.
- Membangun model awal menggunakan Scikit Learn dengan menggunakan data labeled.
- Menggunakan model awal untuk memprediksi label data unlabeled.
- Menambahkan data dengan prediksi label yang memiliki tingkat keyakinan tertentu ke dalam data labeled.
- Membangun kembali model dengan menggunakan data labeled yang sudah ditambahkan dengan data yang telah diprediksi sebelumnya.
- Melakukan tahap 4-6 berulang kali hingga akurasi model tidak lagi meningkat.

Dengan menggunakan teknik Self-Training, kita dapat meningkatkan akurasi model klasifikasi pada dataset Breast Cancer dan memperoleh hasil yang lebih akurat dalam memprediksi apakah sel kanker payudara bersifat jinak atau ganas. Selain itu, dengan menggunakan Seaborn untuk memvisualisasikan tren data, kita dapat dengan mudah mengeksplorasi dataset dan memahami karakteristik dari dataset tersebut.



```
1 from sklearn.semi_supervised import SelfTrainingClassifier
2
3 base_estimator = DecisionTreeClassifier(max_depth=5, random_state=42)
4 self_training_model = SelfTrainingClassifier(base_estimator, max_iter=50, t
hreshold=0.8, verbose=True)
5 self_training_model.fit(X_train, y_train)
6 y_pred = self_training_model.predict(X_test)
7
8
9 print('Accuracy:', metrics.accuracy_score(y_test, y_pred))
10 print('Precision:', metrics.precision_score(y_test, y_pred))
11 print('Recall:', metrics.recall_score(y_test, y_pred))
12 print('F1 Score:', metrics.f1_score(y_test, y_pred))
```

## Output:

```
Accuracy: 0.9181286549707602
Precision: 0.855072463768116
Recall: 0.9365079365079365
F1 Score: 0.8939393939393939
```

## **Kesimpulan**

Dalam laporan teknis ini, kita telah menggunakan Google Collab, Scikit Learn, dan Seaborn Framework untuk menganalisis dataset kanker payudara dan memvisualisasikan tren data. Selain itu, kita juga telah menggunakan decision tree, random forest, dan self-training untuk mengeksplorasi dataset dan membangun model klasifikasi. Dari hasil analisis, dapat disimpulkan bahwa model random forest memiliki akurasi yang lebih baik daripada model decision tree dan self-training. Oleh karena itu, model random forest dapat digunakan untuk memprediksi apakah sel kanker payudara jinak atau ganas.



## Daftar Pustaka:

- <https://scikit-learn.org/stable/>
- <https://seaborn.pydata.org>
- [https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load\\_breast\\_cancer.html](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html)
- <https://www.youtube.com/watch?v=EWThwGSipuY&list=PLiHa1s-EL3vgyJdXmRQExosUuM5IhGBIi>
- [https://scikit-learn.org/stable/auto\\_examples/tree/plot\\_cost\\_complexity\\_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py](https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py)
- [https://scikit-learn.org/stable/auto\\_examples/inspection/plot\\_permutation\\_importance\\_multicollinear.html#sphx-glr-auto-examples-inspection-plot-permutation-importance-multicollinear-py](https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance_multicollinear.html#sphx-glr-auto-examples-inspection-plot-permutation-importance-multicollinear-py)
- [https://scikit-learn.org/stable/auto\\_examples/semi\\_supervised/plot\\_self\\_training\\_varying\\_threshold.html#sphx-glr-auto-examples-semi-supervised-plot-self-training-varying-threshold-py](https://scikit-learn.org/stable/auto_examples/semi_supervised/plot_self_training_varying_threshold.html#sphx-glr-auto-examples-semi-supervised-plot-self-training-varying-threshold-py)
- [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))