

ex8

March 17, 2024

```
[ ]: EX NO 8                               Performance Analysis on Decision Tree Classifier
      DATE - 11/03/2024
      HARIHARAN K - URK22AI1048
```

```
[ ]: '''AIM:

      To Develop a Decision Tree classification model for the Social_Network dataset
      using the scikit-learn.

      DESCRIPTION:

      Decision Tree is the most powerful and popular tool for classification and
      prediction. A Decision tree is a flowchart-like tree structure, where each
      ↪internal
      node denotes a test on an attribute, each branch represents an outcome of the
      test, and each leaf node (terminal node) holds a class label.
```

```
[ ]: #URK22AI1048
      import pandas as pd
      from sklearn.model_selection import train_test_split
      from sklearn.tree import DecisionTreeClassifier
      from sklearn import metrics
      from sklearn.metrics import accuracy_score
      from sklearn.preprocessing import LabelEncoder
      import pandas as pd
      from sklearn.model_selection import train_test_split
      from sklearn.tree import DecisionTreeClassifier
      from sklearn.metrics import confusion_matrix, accuracy_score, recall_score,
      ↪precision_score, f1_score, roc_curve, auc
      from sklearn.preprocessing import LabelEncoder
      import matplotlib.pyplot as plt
      import seaborn as sns

      df=pd.read_csv("Social_Network.csv")
```

```
[ ]: #URK22AI1048
      X = df[['Gender', 'Age', 'EstimatedSalary']]
```

```
y = df['Purchased']
```

```
[ ]: #URK22AI1048
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)
```

```
[ ]: #URK22AI1048
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
↳ random_state=42)
```

```
[ ]: #URK22AI1048
classifier = DecisionTreeClassifier(criterion='entropy')
classifier.fit(X_train, y_train)
```

```
[ ]: DecisionTreeClassifier(criterion='entropy')
```

```
[ ]: #URK22AI1048
y_pred = classifier.predict(X_test)

confusion_mat = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
f_score = f1_score(y_test, y_pred)

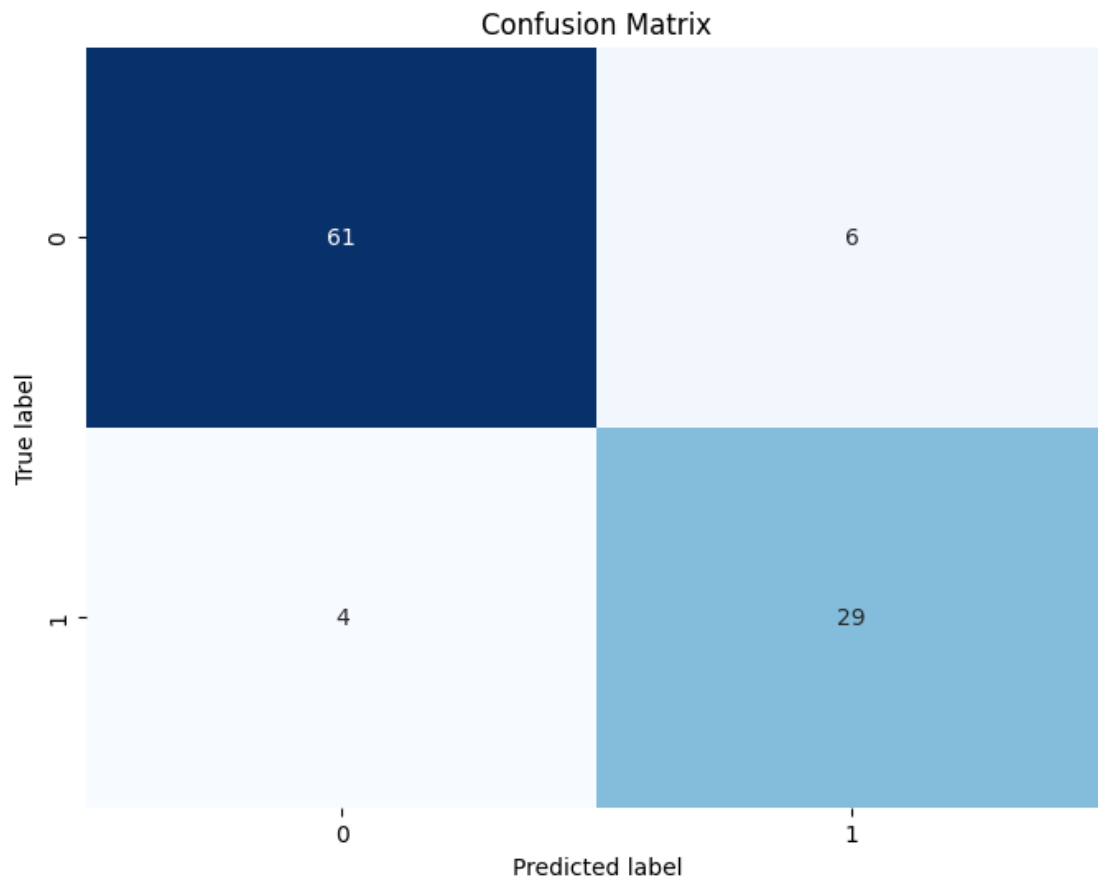
specificity = confusion_mat[0, 0] / (confusion_mat[0, 0] + confusion_mat[0, 1])

fpr, tpr, thresholds = roc_curve(y_test, y_pred)
roc_auc = auc(fpr, tpr)
```

```
[ ]: #URK22AI1048
print('Accuracy:', accuracy)
print('Recall:', recall)
print('Precision:', precision)
print('Specificity:', specificity)
print('F-Score:', f_score)
print('AUC:', roc_auc)
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_mat, annot=True, cmap='Blues', fmt='g', cbar=False)
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Confusion Matrix')
plt.show()
```

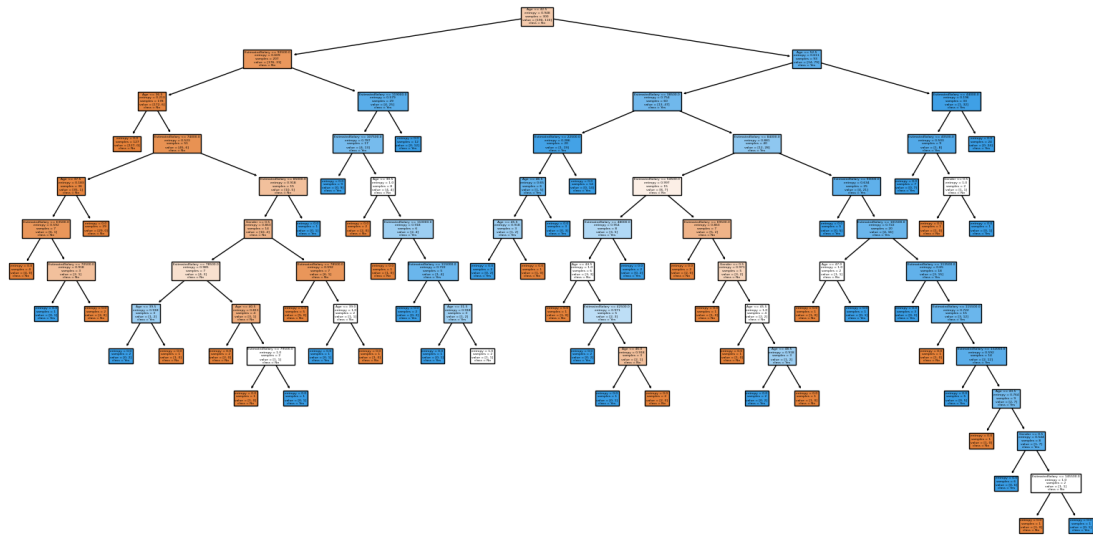
Accuracy: 0.9
Recall: 0.8787878787878788
Precision: 0.8285714285714286

Specificity: 0.9104477611940298
F-Score: 0.8529411764705883
AUC: 0.8946178199909544



```
[ ]: #URK22AI1048
plt.figure(figsize=(20, 10))
from sklearn.tree import plot_tree

class_names = label_encoder.classes_.astype(str)
plot_tree(classifier, feature_names=X.columns, class_names=class_names,
          filled=True)
plt.show()
```



```
[ ]: '''RESULT:
To Develop a Decision Tree classification model for the Social_Network dataset
using the scikit-learn IS executed successfullly.
```