Date Page

5. Decision Trees and Random forest: predict loan approval using a banking dataset.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("loan_data.csv")
print("Dataset preview:")
print(df.head())
df = df.dropna()
cat_cols = df.select_dtypes(include='object').columns
le = LabelEncoder()
for col in cat_cols:
    df[col] = le.fit_transform(df[col])
X = df.drop('loan_status', axis=1)
y = df['loan_status']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
dtree = DecisionTreeClassifier(random_state=42)
dtree.fit(X_train, y_train)
```

rforest = RandomForestClassifier(n_estimators=100, random_state=42)
rforest.fit(X_train, y_train)
dtree_preds = dtree.predict(X_test)
rforest_preds = rforest.predict(X_test)
print("\n ..... Decision Tree performance ....")
print("Confusion Matrix:\n", confusion_matrix(y_test, dtree_preds))
print("Classification Report:\n", classification_report(y_test, dtree_preds))
print("Accuracy Score:", accuracy_score(y_test, dtree_preds))
print("\n ..... Random Forest performance ....")
print("Confusion Matrix:\n", confusion_matrix(y_test, rforest_preds))
print("Classification Report:\n", classification_report(y_test, rforest_preds))
print("Accuracy Score:", accuracy_score(y_test, rforest_preds))
plt.figure(figsize=(20,10))
plot_tree(dtree, feature_names=X.columns, class_names=['Not Approved', 'Approved'], filled=True)
plt.title('Decision Tree Visualization')
plt.show()
importances = rforest.feature_importances_
features = X.columns
feature_importance_df = pd.DataFrame({'Feature': features, 'Importance': importances})
feature_importances_df = feature_importance_df.sort_values(by='Importance',

Date Page

```
ascending = False)
plt.figure(figsize=(12,6))
sns.barplot(x='Importance', y='feature',
    data = feature_importance_df)
plt.title('feature importance from Random
    forest')
plt.show()
```

OUTPUT :

Dataset preview :-

	Loan_ID	Gender	Married	Dependents	Educate	S-E
0	LP001002	Male	No	0	Graduate	No
1	LP001003	Male	Yes	1	Graduate	No
2	LP001005	Male	Yes	0	Graduate	Yes
3	LP001006	Male	Yes	0	Not Graduate	No
4	LP001008	Male	No	0	Graduate	No

	Applicant Income	Coapplicant Income	Loan Amount	Loan_Amount_Term
0	5849	0.0	NaN	360.0
1	4583	1508.0	128.0	360.0
2	3000	0.0	66.0	360.0
3	2583	2358.0	120.0	360.0
4	6000	0.0	141.0	360.0

	Credit_History	Property_Area	Loan_Status
0	1.0	Urban	Y
1	1.0	Rural	N
2	1.0	Urban	Y
3	1.0	Urban	Y
4	1.0	Urban	Y

Date Page

--- Decision Tree performance ---

Confusion Matrix:

[[21 23]
[15 85]]

Classification Report:

	precision	recall	f1-score	support
0	0.58	0.48	0.53	44
1	0.79	0.85	0.82	100
accuracy			0.74	144
macro avg	0.69	0.66	0.67	144
weighted avg	0.72	0.74	0.73	144

Accuracy Score: 0.7361111111111112

--- Random forest performance ---

Confusion matrix:

[[19 25]
[3 97]]

Classification Report:

	precision	recall	f1-score	support
0	0.86	0.43	0.58	44
1	0.80	0.97	0.87	100
accuracy			0.81	144
macro avg	0.83	0.70	0.72	144
weighted avg	0.82	0.81	0.78	144

Accuracy Score: 0.8055555555555556.