Code for exp3

```
# ==========================================
# STEP 1: Import Libraries
# ==========================================

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
from sklearn.preprocessing import LabelEncoder


# ==========================================
# STEP 2: Load Dataset
# ==========================================

df = pd.read_csv("stroke.csv")
print(df.head())


# ==========================================
# STEP 3: Data Preprocessing
# ==========================================

# Drop ID column
df.drop("id", axis=1, inplace=True)

# Fill missing BMI values
df["bmi"].fillna(df["bmi"].mean(), inplace=True)

# Encode categorical columns
le = LabelEncoder()
```

```
cat_cols =
["gender","ever_married","work_type","Residence_type","smoking_status"]

for col in cat_cols:
    df[col] = le.fit_transform(df[col])

print("\nAfter Encoding:")
print(df.head())


# ==========================================
# STEP 4: Define Features and Target
# ==========================================

X = df.drop("stroke", axis=1)
y = df["stroke"]


# ==========================================
# STEP 5: Train-Test Split
# ==========================================

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)


# ==========================================
# STEP 6: DECISION TREE CLASSIFIER
# ==========================================

dt = DecisionTreeClassifier(max_depth=5, random_state=42)
dt.fit(X_train, y_train)

y_pred_dt = dt.predict(X_test)

print("\n===== DECISION TREE RESULTS =====")
print("Accuracy:", accuracy_score(y_test, y_pred_dt))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred_dt))
print("\nClassification Report:\n", classification_report(y_test,
y_pred_dt))


# Plot Confusion Matrix
sns.heatmap(confusion_matrix(y_test, y_pred_dt), annot=True, fmt="d")
plt.title("Decision Tree Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()


# ==========================================
# STEP 7: RANDOM FOREST CLASSIFIER
# ==========================================

rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)

y_pred_rf = rf.predict(X_test)

print("\n===== RANDOM FOREST RESULTS =====")
print("Accuracy:", accuracy_score(y_test, y_pred_rf))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred_rf))
print("\nClassification Report:\n", classification_report(y_test,
y_pred_rf))


# Plot Confusion Matrix
sns.heatmap(confusion_matrix(y_test, y_pred_rf), annot=True, fmt="d")
plt.title("Random Forest Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()


# ==========================================
# STEP 8: MODEL COMPARISON
# ==========================================
```

```python
models = ["Decision Tree", "Random Forest"]
accuracies = [
    accuracy_score(y_test, y_pred_dt),
    accuracy_score(y_test, y_pred_rf)
]

plt.bar(models, accuracies)
plt.title("Model Accuracy Comparison")
plt.ylabel("Accuracy")
plt.show()
```