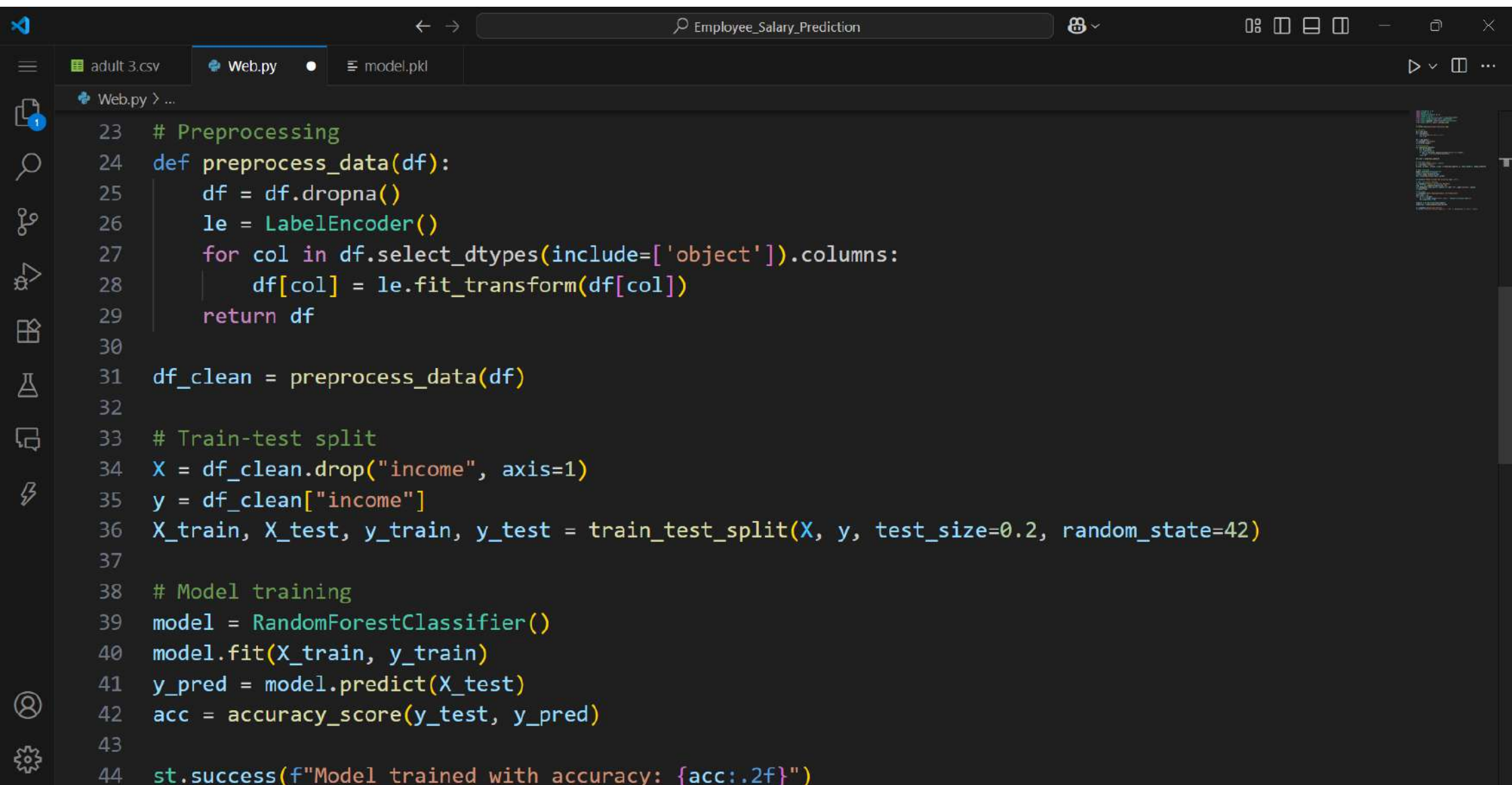


The image shows a Visual Studio Code editor window. The top bar has a search icon, a back/forward arrow, a search bar containing "Employee\_Salary\_Prediction", and a lock icon. Below the top bar, the file explorer shows three files: "adult 3.csv", "Web.py", and "model.pkl". The "Web.py" file is selected and open in the editor. The editor shows the following Python code:

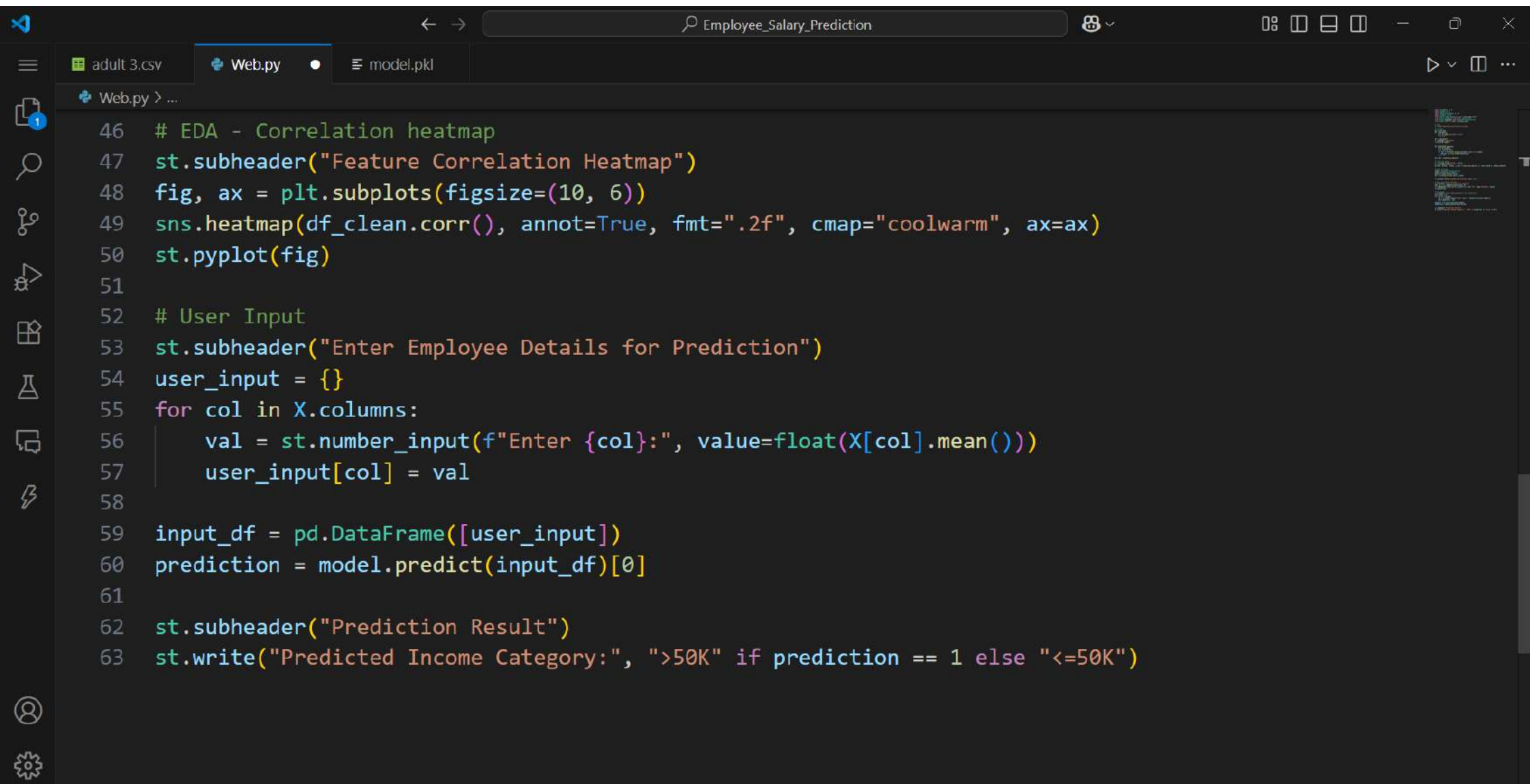
```
1 import streamlit as st
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from sklearn.model_selection import train_test_split
6 from sklearn.preprocessing import LabelEncoder
7 from sklearn.ensemble import RandomForestClassifier
8 from sklearn.metrics import accuracy_score
9
10 # Title
11 st.title("Employee Income Prediction App")
12
13 # Load data
14 @st.cache_data
15 def load_data():
16     df = pd.read_csv("adult 3.csv")
17     return df
18
19 df = load_data()
20 st.subheader("Raw Data")
21 st.write(df.head())
22
```

The code is a Streamlit application for an Employee Income Prediction App. It imports necessary libraries, loads data from a CSV file, and displays the first few rows of the data. The "sklearn.ensemble" module is highlighted in the code.



The image shows a Visual Studio Code editor window with a dark theme. The title bar at the top displays the VS Code logo, navigation arrows, a search bar containing 'Employee\_Salary\_Prediction', and window control buttons. The Explorer sidebar on the left shows a file tree with 'adult 3.csv', 'Web.py', and 'model.pkl'. The 'Web.py' file is open in the main editor. The code in the editor is a Python script for preprocessing data, splitting it into training and testing sets, training a Random Forest Classifier, and printing the accuracy. The code is as follows:

```
23 # Preprocessing
24 def preprocess_data(df):
25     df = df.dropna()
26     le = LabelEncoder()
27     for col in df.select_dtypes(include=['object']).columns:
28         df[col] = le.fit_transform(df[col])
29     return df
30
31 df_clean = preprocess_data(df)
32
33 # Train-test split
34 X = df_clean.drop("income", axis=1)
35 y = df_clean["income"]
36 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
37
38 # Model training
39 model = RandomForestClassifier()
40 model.fit(X_train, y_train)
41 y_pred = model.predict(X_test)
42 acc = accuracy_score(y_test, y_pred)
43
44 st.success(f"Model trained with accuracy: {acc:.2f}")
```



The image shows a Visual Studio Code editor window. The title bar at the top reads "Employee\_Salary\_Prediction". The file explorer on the left shows three files: "adult 3.csv", "Web.py", and "model.pkl". The "Web.py" file is currently open and displays the following Python code:

```
46 # EDA - Correlation heatmap
47 st.subheader("Feature Correlation Heatmap")
48 fig, ax = plt.subplots(figsize=(10, 6))
49 sns.heatmap(df_clean.corr(), annot=True, fmt=".2f", cmap="coolwarm", ax=ax)
50 st.pyplot(fig)
51
52 # User Input
53 st.subheader("Enter Employee Details for Prediction")
54 user_input = {}
55 for col in X.columns:
56     val = st.number_input(f"Enter {col}:", value=float(X[col].mean()))
57     user_input[col] = val
58
59 input_df = pd.DataFrame([user_input])
60 prediction = model.predict(input_df)[0]
61
62 st.subheader("Prediction Result")
63 st.write("Predicted Income Category:", ">50K" if prediction == 1 else "<=50K")
```