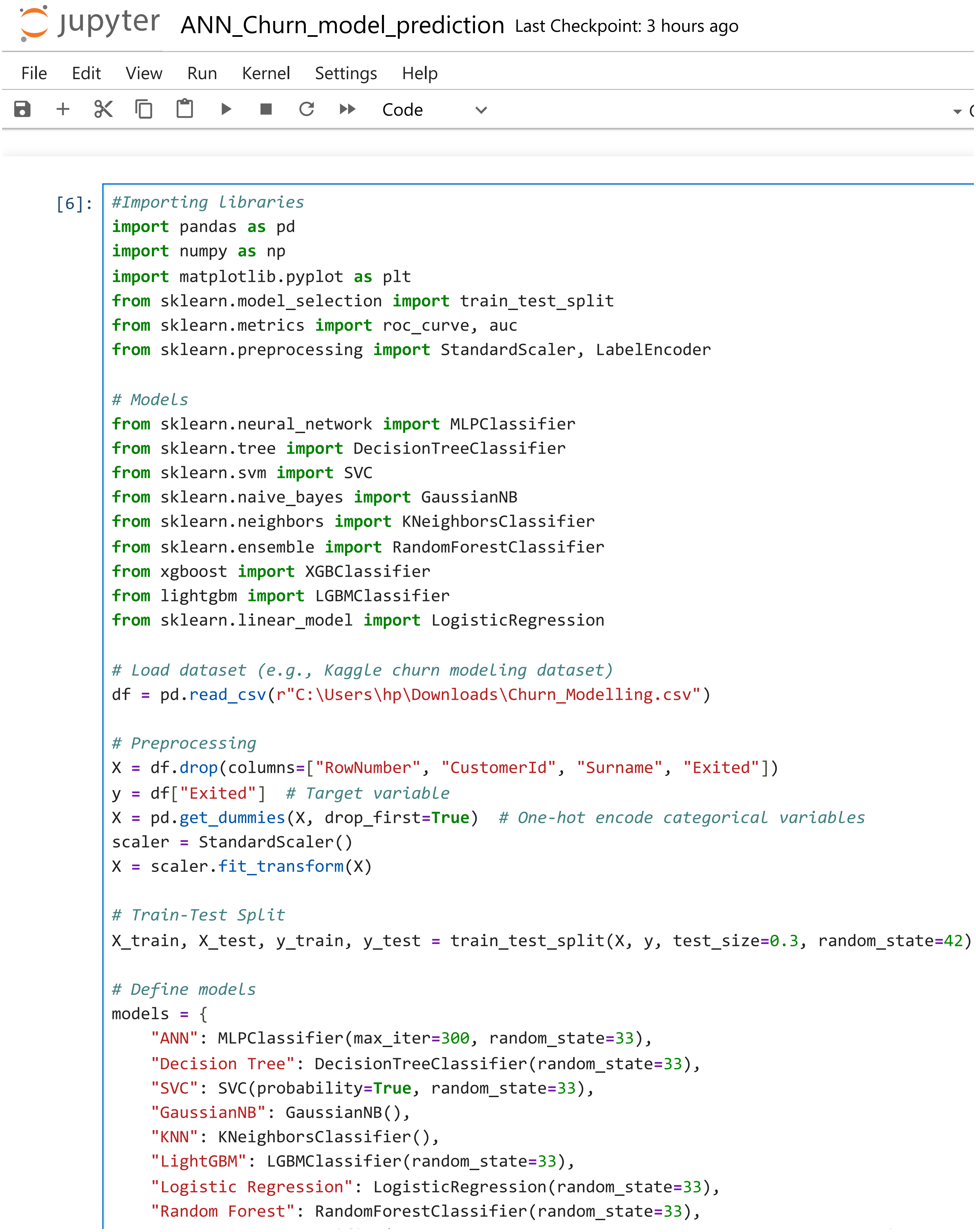

```
#Importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Models
from sklearn.neural_network import MLPClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from sklearn.linear_model import LogisticRegression

# Load dataset (e.g., Kaggle churn modeling dataset)
df = pd.read_csv(r"C:\Users\hp\Downloads\Churn_Modelling.csv")

# Preprocessing
X = df.drop(columns=["RowNumber", "CustomerId", "Surname", "Exited"])
y = df["Exited"]  # Target variable
X = pd.get_dummies(X, drop_first=True)  # One-hot encode categorical variables
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Define models
models = {
    "ANN": MLPClassifier(max_iter=300, random_state=33),
    "Decision Tree": DecisionTreeClassifier(random_state=33),
    "SVC": SVC(probability=True, random_state=33),
    "GaussianNB": GaussianNB(),
    "KNN": KNeighborsClassifier(),
    "LightGBM": LGBMClassifier(random_state=33),
    "Logistic Regression": LogisticRegression(random_state=33),
    "Random Forest": RandomForestClassifier(random_state=33),
```