```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "932bfd61",
   "metadata": {},
   "source": [
    "# ðŸ“Š Customer Churn Prediction using Machine Learning\n",
    "\n",
    "**Goal**: Predict whether a customer will churn (leave the company) based on service usage.\n",
    "\n",
    "This project demonstrates a full machine learning pipeline: EDA, data cleaning, modeling, and evaluation.\n",
    "\n",
    "Dataset used: [Telco Customer Churn - Kaggle](https://www.kaggle.com/blastchar/telco-customer-churn)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "61fbfa5d",
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd\n",
    "import numpy as np\n",
    "import matplotlib.pyplot as plt\n",
    "import seaborn as sns\n",
    "\n",
    "from sklearn.model_selection import train_test_split\n",
    "from sklearn.preprocessing import StandardScaler\n",
    "from sklearn.ensemble import RandomForestClassifier\n",
    "from sklearn.metrics import classification_report, confusion_matrix, accuracy_score"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "947f2cd5",
   "metadata": {},
   "outputs": [],
   "source": [
    "# Load dataset\n",
    "df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')\n",
    "df.head()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "feebbb87",
   "metadata": {},
   "outputs": [],
   "source": [
    "# Data Cleaning\n",
    "df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')\n",
    "df['TotalCharges'].fillna(df['TotalCharges'].median(), inplace=True)\n",
    "df.drop('customerID', axis=1, inplace=True)"
```

```
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "94caa990",
   "metadata": {},
   "outputs": [],
   "source": [
    "# Encode categorical columns\n",
    "df['Churn'] = df['Churn'].map({'Yes': 1, 'No': 0})\n",
    "df = pd.get_dummies(df, drop_first=True)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "316f4d29",
   "metadata": {},
   "outputs": [],
   "source": [
    "# Train-Test Split\n",
    "X = df.drop('Churn', axis=1)\n",
    "y = df['Churn']\n",
    "X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "51b0ab00",
   "metadata": {},
   "outputs": [],
   "source": [
    "# Feature Scaling\n",
    "scaler = StandardScaler()\n",
    "X_train = scaler.fit_transform(X_train)\n",
    "X_test = scaler.transform(X_test)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "bc5cc3a8",
   "metadata": {},
   "outputs": [],
   "source": [
    "# Train Random Forest Classifier\n",
    "model = RandomForestClassifier(n_estimators=100, random_state=42)\n",
    "model.fit(X_train, y_train)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "56ba3c85",
   "metadata": {},
   "outputs": [],
   "source": [
    "# Evaluate Model\n",
```

```
        "y_pred = model.predict(X_test)\n",
        "print(\"Accuracy:\", accuracy_score(y_test, y_pred))\n",
        "print(classification_report(y_test, y_pred))\n",
        "sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')\n",
        "plt.title(\"Confusion Matrix\")\n",
        "plt.show()"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": null,
       "id": "ee7c2ac6",
       "metadata": {},
       "outputs": [],
       "source": [
        "# Feature Importance\n",
        "feat_imp = pd.Series(model.feature_importances_, index=X.columns)\n",
        "feat_imp.nlargest(10).plot(kind='barh')\n",
        "plt.title(\"Top 10 Feature Importances\")\n",
        "plt.show()"
       ]
      }
     ],
     "metadata": {},
     "nbformat": 4,
     "nbformat_minor": 5
    }
```