```
! pip install ydata-profiling
```

```
Collecting ydata-profiling
  Downloading ydata_profiling-4.16.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting scipy<1.16,>=1.4.1 (from ydata-profiling)
  Downloading scipy-1.15.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (61 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.0/62.0 kB 2.3 MB/s eta 0:00:00
Requirement already satisfied: pandas!=1.4.0,<3.0,>1.1 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (2.2.2)
Requirement already satisfied: matplotlib<=3.10,>=3.5 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (3.10.0)
Requirement already satisfied: pydantic>=2 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (2.11.7)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (6.0.2)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (3.1.6)
Collecting visions<0.8.2,>=0.7.5 (from visions[type_image_path]<0.8.2,>=0.7.5->ydata-profiling)
  Downloading visions-0.8.1-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.2,>=1.16.0 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (2.0.2)
Collecting htmlmin==0.1.12 (from ydata-profiling)
  Downloading htmlmin-0.1.12.tar.gz (19 kB)
  Preparing metadata (setup.py) ... done
Collecting phik<0.13,>=0.11.1 (from ydata-profiling)
  Downloading phik-0.12.5-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl.metadata (5.6 kB)
Requirement already satisfied: requests<3,>=2.24.0 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (2.32.4)
Requirement already satisfied: tqdm<5,>=4.48.2 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (4.67.1)
Requirement already satisfied: seaborn<0.14,>=0.10.1 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (0.13.2)
Collecting multimethod<2,>=1.4 (from ydata-profiling)
  Downloading multimethod-1.12-py3-none-any.whl.metadata (9.6 kB)
Requirement already satisfied: statsmodels<1,>=0.13.2 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (0.14.5)
Requirement already satisfied: typeguard<5,>=3 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (4.4.4)
Collecting imagehash==4.3.1 (from ydata-profiling)
  Downloading ImageHash-4.3.1-py2.py3-none-any.whl.metadata (8.0 kB)
Requirement already satisfied: wordcloud>=1.9.3 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (1.9.4)
Collecting dacite>=1.8 (from ydata-profiling)
  Downloading dacite-1.9.2-py3-none-any.whl.metadata (17 kB)
Requirement already satisfied: numba<=0.61,>=0.56.0 in /usr/local/lib/python3.12/dist-packages (from ydata-profiling) (0.60.0)
Requirement already satisfied: PyWavelets in /usr/local/lib/python3.12/dist-packages (from imagehash==4.3.1->ydata-profiling) (1.
Requirement already satisfied: pillow in /usr/local/lib/python3.12/dist-packages (from imagehash==4.3.1->ydata-profiling) (11.3.0
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2<3.2,>=2.11.1->ydata-profil
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib<=3.10,>=3.5->ydata-pr
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib<=3.10,>=3.5->ydata-profil
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib<=3.10,>=3.5->ydata-p
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib<=3.10,>=3.5->ydata-p
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib<=3.10,>=3.5->ydata-pro
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib<=3.10,>=3.5->ydata-pr
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib<=3.10,>=3.5->ydat
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.12/dist-packages (from numba<=0.61,>=0.56.0->
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas!=1.4.0,<3.0,>1.1->ydata-profi
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas!=1.4.0,<3.0,>1.1->ydata-pro
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.12/dist-packages (from phik<0.13,>=0.11.1->ydata-profilin
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2->ydata-profili
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2->ydata-profilin
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2->ydata-prof
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from pydantic>=2->ydata-profi
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3,>=2.24.0->yda
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3,>=2.24.0->ydata-profiling
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3,>=2.24.0->ydata-pro
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3,>=2.24.0->ydata-pro
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.12/dist-packages (from statsmodels<1,>=0.13.2->ydata-profil
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.12/dist-packages (from visions<0.8.2,>=0.7.5->visions[type_
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.12/dist-packages (from visions<0.8.2,>=0.7.5->visions[type_
Collecting puremagic (from visions<0.8.2,>=0.7.5->visions[type_image_path]<0.8.2,>=0.7.5->ydata-profiling)
  Downloading puremagic-1.30-py3-none-any.whl.metadata (5.8 kB)
```

```
import pandas as pd
from ydata_profiling import ProfileReport
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
```

Upgrade to ydata-sdk

Improve your data and profiling with ydata-sdk, featuring data quality scoring, redundancy detection, outlier identification, text validation, and synthetic data generation.

```
df = pd.read_csv('Telechurn.csv')
df.head(10)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	..
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	..
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	..
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	..
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	..
5	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	..
6	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	..
7	6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL	Yes	..
8	7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	..
9	6388-TABGU	Male	0	No	Yes	62	Yes	No	DSL	Yes	..

10 rows × 21 columns

```
# Generate EDA report
profile = ProfileReport(df, title="Telco Customer Churn EDA", explorative=True)
profile.to_file("Telechurn.html")  # Display in html
```

```
Summarize dataset: 100%                                34/34 [00:03<00:00,   4.91it/s, Completed]

  0%|          | 0/21 [00:00<?, ?it/s]
  5%|▌         | 1/21 [00:00<00:04,  4.27it/s]
 48%|████▊     | 10/21 [00:00<00:00, 28.38it/s]
 67%|██████▋   | 14/21 [00:00<00:00, 29.21it/s]
100%|██████████| 21/21 [00:00<00:00, 26.59it/s]
Generate report structure: 100%                                1/1 [00:05<00:00,   5.46s/it]

Render HTML: 100%                                1/1 [00:00<00:00,   1.18it/s]

Export report to file: 100%                                1/1 [00:00<00:00, 41.38it/s]
```

```
# Basic preprocessing from EDA insights
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')  # Convert to numeric, handle blanks
df = df.dropna()  # Drop rows with NaN in TotalCharges (11 rows)
df = df.drop('customerID', axis=1)  # Irrelevant
```

```
# Prepare features and target
X = df.drop('Churn', axis=1)
y = df['Churn'].map({'Yes': 1, 'No': 0})  # Encode target
```

```
# Identify categorical and numeric columns
numeric_features = ['tenure', 'MonthlyCharges', 'TotalCharges']
categorical_features = [col for col in X.columns if col not in numeric_features]
```

```
# Preprocessing pipeline
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numeric_features),
        ('cat', OneHotEncoder(drop='first', handle_unknown='ignore'), categorical_features)
    ])
```

```
# Full pipeline with logistic regression
pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', LogisticRegression(max_iter=1000))
])
```

```
# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Train and predict
pipeline.fit(X_train, y_train)
y_pred = pipeline.predict(X_test)
```

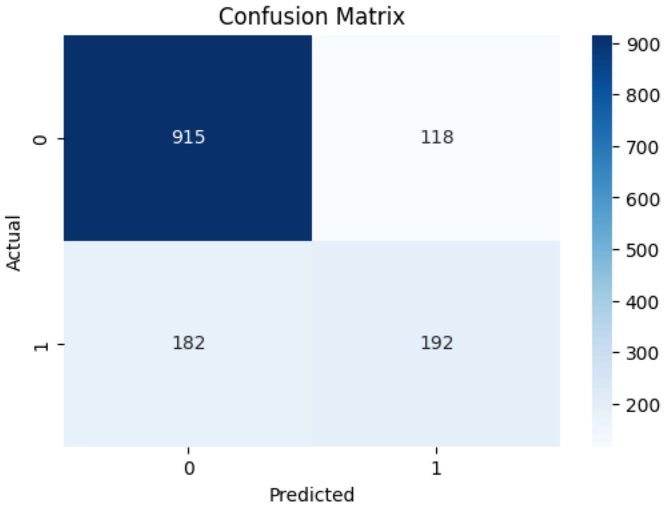
```
# Evaluate
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.79
              precision    recall  f1-score   support

           0       0.83      0.89      0.86      1033
           1       0.62      0.51      0.56       374

    accuracy                           0.79      1407
   macro avg       0.73      0.70      0.71      1407
weighted avg       0.78      0.79      0.78      1407
```

```
#Plot confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```



```
from google.colab import files
```

```
# Add predictions to test set
test_df = X_test.copy()
test_df['Actual_Churn'] = y_test
test_df['Predicted_Churn'] = y_pred

# Export
test_df.to_csv('telco_churn_predictions.csv', index=False)
files.download('telco_churn_predictions.csv')  # Downloads to your local machine
```

```
Start coding or generate with AI.
```

Start coding or generate with AI.

Start coding or generate with AI.