In [1]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\ayush\anaconda3\lib\site-packages (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\ayush\anaconda3\lib\site-packages (from pandas) (2.
8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\ayush\anaconda3\lib\site-packages (from pandas) (2023.3.post
1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\ayush\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\ayush\anaconda3\lib\site-packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\ayush\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->p
andas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import pandas as pd
```

In [5]:
```
df = pd.read_csv('data.csv')
```

In [6]:
```
df
```

Out[6]:

	Unnamed: 0	id	cons_12m	cons_gas_12m	cons_last_month	forecast_cons_12m	forecast_discount_energy
0	0	24011ae4ebbe3035111d65fa7c15bc57	0.000000	4.739944	0.000000	0.000000	0.0
1	1	d29c2c54acc38ff3c0614d0a653813dd	3.668479	0.000000	0.000000	2.280920	0.0
2	2	764c75f661154dac3a6c254cd082ea7d	2.736397	0.000000	0.000000	1.689841	0.0
3	3	bba03439a292a1e166f80264c16191cb	3.200029	0.000000	0.000000	2.382089	0.0
4	4	149d57cf92fc41cf94415803a877cb4b	3.646011	0.000000	2.721811	2.650065	0.0
...	...	...	...	...	...	...	...
14601	14601	18463073fb097fc0ac5d3e040f356987	4.508812	4.680707	0.000000	3.667360	0.0
14602	14602	d0a6f71671571ed83b2645d23af6de00	3.858778	0.000000	2.260071	2.801191	0.0
14603	14603	10e6828ddd62cbcf687cb74928c4c2d2	3.265996	0.000000	2.255273	2.281919	0.0
14604	14604	1cf20fd6206d7678d5bcafd28c53b4db	2.120574	0.000000	0.000000	1.308351	0.0
14605	14605	563dde550fd624d7352f3de77c0cdfcd	3.941064	0.000000	0.000000	2.882758	0.0

14606 rows × 64 columns

In [7]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
```

In [8]:
```
df = pd.read_csv('data.csv')
```

In [13]:
```
print("Dataset Overview:")
print(df.head())
```

```
Dataset Overview:
   Unnamed: 0                                id  cons_12m  cons_gas_12m  \
0           0  24011ae4ebbe3035111d65fa7c15bc57  0.000000      4.739944
1           1  d29c2c54acc38ff3c0614d0a653813dd  3.668479      0.000000
2           2  764c75f661154dac3a6c254cd082ea7d  2.736397      0.000000
3           3  bba03439a292a1e166f80264c16191cb  3.200029      0.000000
4           4  149d57cf92fc41cf94415803a877cb4b  3.646011      0.000000

   cons_last_month  forecast_cons_12m  forecast_discount_energy  \
0         0.000000           0.000000                       0.0
1         0.000000           2.280920                       0.0
2         0.000000           1.689841                       0.0
3         0.000000           2.382089                       0.0
4         2.721811           2.650065                       0.0

   forecast_meter_rent_12m  forecast_price_energy_off_peak  \
0                 0.444045                        0.114481
1                 1.237292                        0.145711
2                 1.599009                        0.165794
3                 1.318689                        0.146694
4                 2.122969                        0.116900

   forecast_price_energy_peak  ...  months_modif_prod  months_renewal  \
0                    0.098142  ...                  2               6
1                    0.000000  ...                 76               4
2                    0.087899  ...                 68               8
3                    0.000000  ...                 69               9
4                    0.100015  ...                 71               9

   channel_MISSING  channel_ewpakwlliwisiwduibdlfmalxowmwpci  \
0                0                                         0
1                1                                         0
2                0                                         0
3                0                                         0
4                1                                         0

   channel_foosdfpfkusacimwkcsosbicdxkicaua  \
0                                         1
1                                         0
2                                         1
3                                         0
```

```
4                                          0

   channel_lmkebamcaaclubfxadlmueccxoimlema  \
0                                          0
1                                          0
2                                          0
3                                          1
4                                          0

   channel_usilxuppasemubllopkaafesmlibmsdf  \
0                                          0
1                                          0
2                                          0
3                                          0
4                                          0

   origin_up_kamkkxfxxuwbdslkwifmmcsiusiuosws  \
0                                           0
1                                           1
2                                           1
3                                           1
4                                           1

   origin_up_ldkssxwpmemidmecebumciepifcamkci  \
0                                           0
1                                           0
2                                           0
3                                           0
4                                           0

   origin_up_lxidpiddsbxsbosboudacockeimpuepw
0                                           1
1                                           0
2                                           0
3                                           0
4                                           0

[5 rows x 64 columns]
```

In [20]:
```
df_encoded = pd.get_dummies(df)
```

In [21]:
```
X = df.drop("churn", axis=1)
y = df["churn"]
```

In [22]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [28]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
```

In [37]:
```
label_encoder = LabelEncoder()
df['your_categorical_column_encoded'] = label_encoder.fit_transform(df['id'])
```

In [65]:
```
X = df.drop(["churn", 'id'], axis=1)
y = df["churn"]
```

In [64]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [61]:
```
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
```

Out[61]:
```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

In [62]:
```
y_pred = rf_model.predict(X_test)
```

In [63]:
```
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 0.90
```

In [59]:
```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
```

In [68]:
```
print("Evaluation Metrics:")
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1 Score: {f1:.2f}")
print("Confusion Matrix:")
print(conf_matrix)
```

```
Evaluation Metrics:
Accuracy: 0.90
Precision: 0.76
Recall: 0.04
F1 Score: 0.08
Confusion Matrix:
[[2613    4]
 [ 292   13]]
```

In [ ]: