

Credit Card Analysis

Importing Libraries

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report,confusion_matrix,roc_auc_score, r
from sklearn.preprocessing import StandardScaler
from sklearn.utils import resample
```

Loading Data

```
In [2]: df=pd.read_csv("C:\\\\Users\\\\byash\\\\Finance Fraud Prediction (python)\\\\creditcard.csv")
```

```
In [3]: df
```

```
Out[3]:
```

	Time	V1	V2	V3	V4	V5	V6
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921
...
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617

284807 rows × 31 columns



Data Information And Data cleaning

```
In [4]: df.columns
```

```
Out[4]: Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',  
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',  
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',  
       'Class'],  
       dtype='object')
```

```
In [5]: df.head()
```

```
Out[5]:
```

	Time	V1	V2	V3	V4	V5	V6	V7	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.0986
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.0851
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.2476
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.3774
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.2705

5 rows × 31 columns



```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count  Dtype  
---  -- 
 0   Time     284807 non-null   float64
 1   V1       284807 non-null   float64
 2   V2       284807 non-null   float64
 3   V3       284807 non-null   float64
 4   V4       284807 non-null   float64
 5   V5       284807 non-null   float64
 6   V6       284807 non-null   float64
 7   V7       284807 non-null   float64
 8   V8       284807 non-null   float64
 9   V9       284807 non-null   float64
 10  V10      284807 non-null   float64
 11  V11      284807 non-null   float64
 12  V12      284807 non-null   float64
 13  V13      284807 non-null   float64
 14  V14      284807 non-null   float64
 15  V15      284807 non-null   float64
 16  V16      284807 non-null   float64
 17  V17      284807 non-null   float64
 18  V18      284807 non-null   float64
 19  V19      284807 non-null   float64
 20  V20      284807 non-null   float64
 21  V21      284807 non-null   float64
 22  V22      284807 non-null   float64
 23  V23      284807 non-null   float64
 24  V24      284807 non-null   float64
 25  V25      284807 non-null   float64
 26  V26      284807 non-null   float64
 27  V27      284807 non-null   float64
 28  V28      284807 non-null   float64
 29  Amount    284807 non-null   float64
 30  Class     284807 non-null   int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

In [7]: `df.describe()`

Out[7]:

	Time	V1	V2	V3	V4	
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	1.168375e-15	3.416908e-16	-1.379537e-15	2.074095e-15	9.6040
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.3802
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.1374
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.9159
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.4335
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.1192
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.4801

8 rows × 31 columns



In [8]: `df.isnull().sum()`

Out[8]:

Time	0
V1	0
V2	0
V3	0
V4	0
V5	0
V6	0
V7	0
V8	0
V9	0
V10	0
V11	0
V12	0
V13	0
V14	0
V15	0
V16	0
V17	0
V18	0
V19	0
V20	0
V21	0
V22	0
V23	0
V24	0
V25	0
V26	0
V27	0
V28	0
Amount	0
Class	0
	dtype: int64

```
In [9]: df.duplicated()
```

```
Out[9]: 0      False
1      False
2      False
3      False
4      False
...
284802  False
284803  False
284804  False
284805  False
284806  False
Length: 284807, dtype: bool
```

```
In [10]: df_1=df.duplicated().sum()
df_1
```

```
Out[10]: 1081
```

```
In [11]: df.drop_duplicates()
```

```
Out[11]:    Time      V1      V2      V3      V4      V5      V6
0      0.0  -1.359807 -0.072781  2.536347  1.378155 -0.338321  0.462388  0.2395
1      0.0   1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361 -0.0786
2      1.0  -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499  0.7914
3      1.0  -0.966272 -0.185226  1.792993 -0.863291 -0.010309  1.247203  0.2376
4      2.0  -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921  0.5929
...
284802 172786.0 -11.881118 10.071785 -9.834783 -2.066656 -5.364473 -2.606837 -4.9182
284803 172787.0 -0.732789 -0.055080  2.035030 -0.738589  0.868229  1.058415  0.0243
284804 172788.0  1.919565 -0.301254 -3.249640 -0.557828  2.630515  3.031260 -0.2968
284805 172788.0 -0.240440  0.530483  0.702510  0.689799 -0.377961  0.623708 -0.6861
284806 172792.0 -0.533413 -0.189733  0.703337 -0.506271 -0.012546 -0.649617  1.5770
```

283726 rows × 31 columns



EDA(Exploratory Data Analysis)

```
In [12]: unique_classes = df['Class'].unique()
unique_classes
df['Class'].value_counts()
```

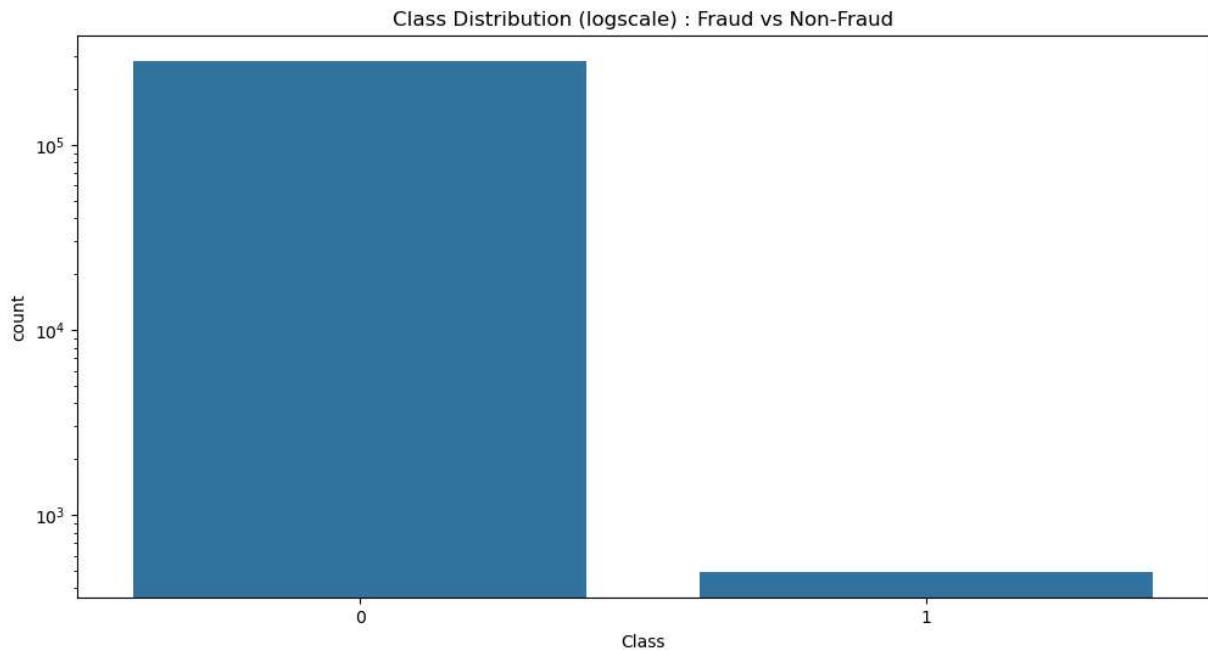
```
Out[12]: Class
0    284315
1     492
Name: count, dtype: int64
```

```
In [13]: correlation_matrix = df.corr()
print(correlation_matrix['Class'].sort_values(ascending=False))
```

```
Class      1.000000
V11      0.154876
V4       0.133447
V2       0.091289
V21      0.040413
V19      0.034783
V20      0.020090
V8       0.019875
V27      0.017580
V28      0.009536
Amount    0.005632
V26      0.004455
V25      0.003308
V22      0.000805
V23     -0.002685
V15     -0.004223
V13     -0.004570
V24     -0.007221
Time     -0.012323
V6      -0.043643
V5      -0.094974
V9      -0.097733
V1      -0.101347
V18     -0.111485
V7      -0.187257
V3      -0.192961
V16     -0.196539
V10     -0.216883
V12     -0.260593
V14     -0.302544
V17     -0.326481
Name: Class, dtype: float64
```

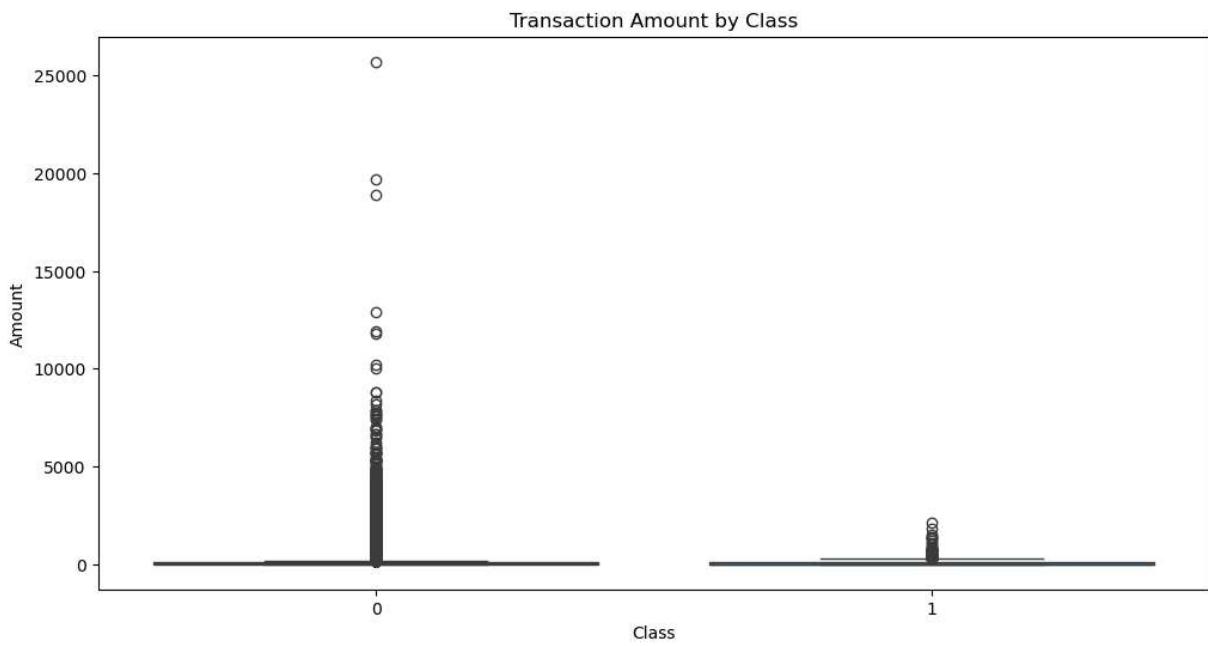
Class Distribution : Fraud vs Non-Fraud'

```
In [14]: plt.figure(figsize=(12, 6))
sns.countplot(x='Class', data=df)
plt.yscale('log')
plt.title('Class Distribution (logscale) : Fraud vs Non-Fraud')
plt.show()
```



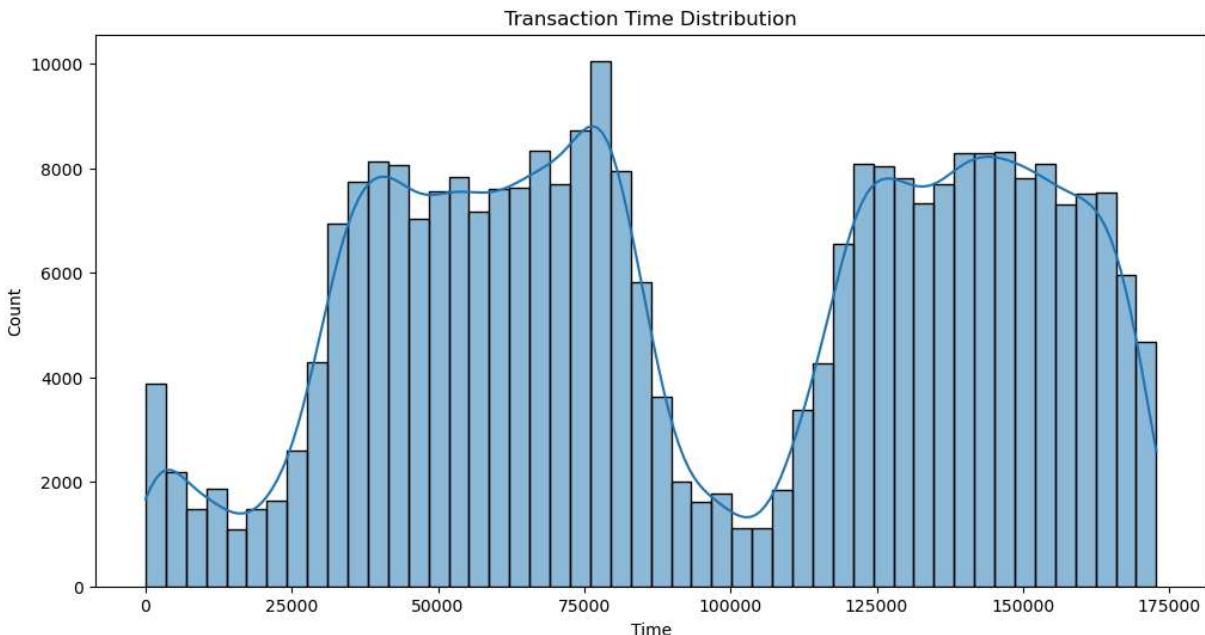
Amount distribution by class

```
In [15]: plt.figure(figsize=(12, 6))
sns.boxplot(x='Class', y='Amount', data=df)
plt.title('Transaction Amount by Class')
plt.show()
```



Time distribution

```
In [16]: plt.figure(figsize=(12, 6))
sns.histplot(df['Time'], bins=50, kde=True)
plt.title('Transaction Time Distribution')
plt.show()
```



Machine Learning - Logistic Regression

```
In [17]: X = df.drop('Class', axis=1)
y = df['Class']
```

scale the Amount and Time features

```
In [18]: scaler = StandardScaler()
X['Amount'] = scaler.fit_transform(X['Amount'].values.reshape(-1, 1))
X['Time'] = scaler.fit_transform(X['Time'].values.reshape(-1, 1))
```

Split data into training and testing sets

```
In [19]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
In [20]: train_data = pd.concat([X_train, y_train], axis=1)
```

Separate majority and minority classes

```
In [21]: majority = train_data[train_data['Class'] == 0]
minority = train_data[train_data['Class'] == 1]
```

```
In [22]: minority_upsampled = resample(minority,
                                     replace=True,
                                     n_samples=len(majority),
                                     random_state=42)
train_upsampled = pd.concat([majority, minority_upsampled])
```

Separate features and target

```
In [23]: X_train_upsampled = train_upsampled.drop('Class', axis=1)
y_train_upsampled = train_upsampled['Class']
```

Train logistic regression model

```
In [24]: model = LogisticRegression(max_iter=1000)
model.fit(X_train_upsampled, y_train_upsampled)
```

```
Out[24]: LogisticRegression(max_iter=1000)
```

Make predictions

```
In [25]: y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)[:, 1]
```

```
In [26]: y_pred
```

```
Out[26]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

```
In [27]: y_proba
```

```
Out[27]: array([0.00076901, 0.06098385, 0.06339756, ..., 0.0039372 , 0.06982748,
0.00721471])
```

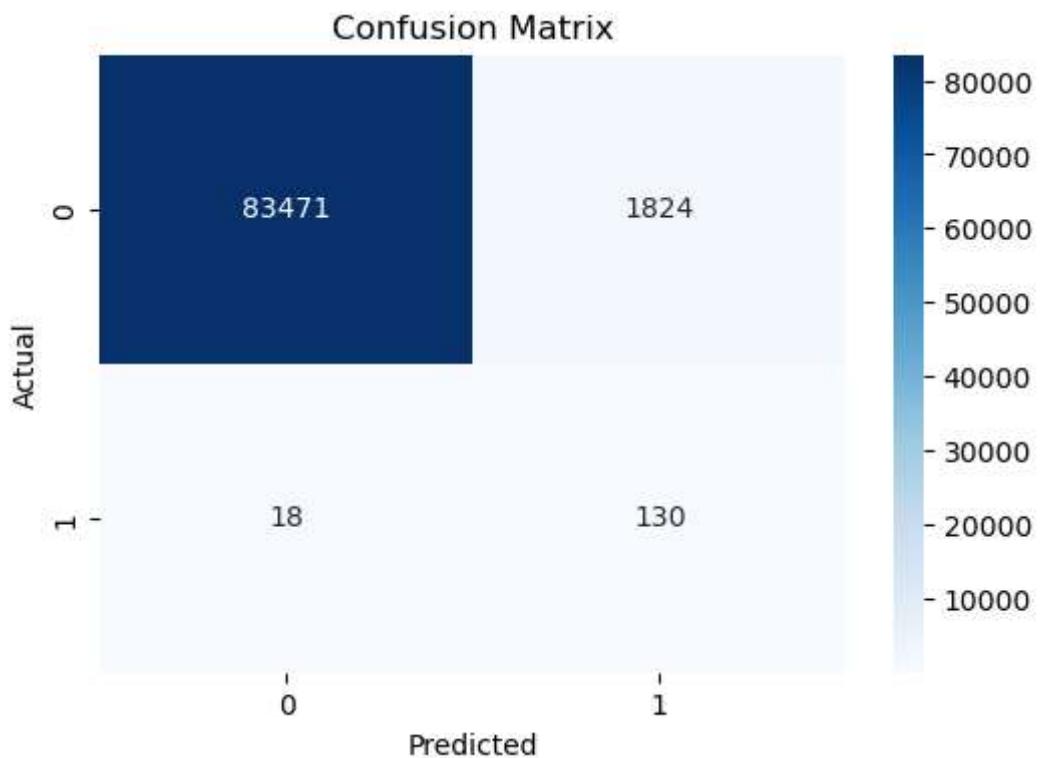
```
In [28]: ### Evaluation metrics and Confusion matrix
```

Evaluation metrics & Confusion matrix

```
In [29]: # Evaluation metrics
print(classification_report(y_test, y_pred))

# 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()
```

	precision	recall	f1-score	support
0	1.00	0.98	0.99	85295
1	0.07	0.88	0.12	148
accuracy			0.98	85443
macro avg	0.53	0.93	0.56	85443
weighted avg	1.00	0.98	0.99	85443



```
In [31]: pip install nbconvert
```

Requirement already satisfied: nbconvert in c:\users\byash\anaconda3\lib\site-packages (7.16.4)
Requirement already satisfied: beautifulsoup4 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (4.12.3)
Requirement already satisfied: bleach!=5.0.0 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (4.1.0)
Requirement already satisfied: defusedxml in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (0.7.1)
Requirement already satisfied: jinja2>=3.0 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (3.1.4)
Requirement already satisfied: jupyter-core>=4.7 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (5.7.2)
Requirement already satisfied: jupyterlab-pygments in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (0.1.2)
Requirement already satisfied: markupsafe>=2.0 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (2.1.3)
Requirement already satisfied: mistune<4,>=2.0.3 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (2.0.4)
Requirement already satisfied: nbclient>=0.5.0 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (0.8.0)
Requirement already satisfied: nbformat>=5.7 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (5.10.4)
Requirement already satisfied: packaging in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (24.1)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (1.5.0)
Requirement already satisfied: pygments>=2.4.1 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (2.15.1)
Requirement already satisfied: tinycss2 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (1.2.1)
Requirement already satisfied: traitlets>=5.1 in c:\users\byash\anaconda3\lib\site-packages (from nbconvert) (5.14.3)
Requirement already satisfied: six>=1.9.0 in c:\users\byash\anaconda3\lib\site-packages (from bleach!=5.0.0->nbconvert) (1.16.0)
Requirement already satisfied: webencodings in c:\users\byash\anaconda3\lib\site-packages (from bleach!=5.0.0->nbconvert) (0.5.1)
Requirement already satisfied: platformdirs>=2.5 in c:\users\byash\anaconda3\lib\site-packages (from jupyter-core>=4.7->nbconvert) (3.10.0)
Requirement already satisfied: pywin32>=300 in c:\users\byash\anaconda3\lib\site-packages (from jupyter-core>=4.7->nbconvert) (305.1)
Requirement already satisfied: jupyter-client>=6.1.12 in c:\users\byash\anaconda3\lib\site-packages (from nbclient>=0.5.0->nbconvert) (8.6.0)
Requirement already satisfied: fastjsonschema>=2.15 in c:\users\byash\anaconda3\lib\site-packages (from nbformat>=5.7->nbconvert) (2.16.2)
Requirement already satisfied: jsonschema>=2.6 in c:\users\byash\anaconda3\lib\site-packages (from nbformat>=5.7->nbconvert) (4.23.0)
Requirement already satisfied: soupsieve>1.2 in c:\users\byash\anaconda3\lib\site-packages (from beautifulsoup4->nbconvert) (2.5)
Requirement already satisfied: attrs>=22.2.0 in c:\users\byash\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert) (23.1.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in c:\users\byash\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert) (2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in c:\users\byash\anaconda3\lib\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in c:\users\byash\anaconda3\lib\site-p

```
ackages (from jsonschema>=2.6->nbformat>=5.7->nbconvert) (0.10.6)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\byash\anaconda3\lib\site-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (2.9.0.post0)
Requirement already satisfied: pyzmq>=23.0 in c:\users\byash\anaconda3\lib\site-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (25.1.2)
Requirement already satisfied: tornado>=6.2 in c:\users\byash\anaconda3\lib\site-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (6.4.1)
Note: you may need to restart the kernel to use updated packages.
```

In []: