

```
# Basic libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# ML Libraries
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
```

```
from google.colab import files
uploaded = files.upload()
```



Browse...

No files selected.

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable

Start coding or [generate](#) with AI.

```
from google.colab import files
uploaded = files.upload()
```



Browse...

cs-training.csv

cs-training.csv(application/vnd.ms-excel) - 7564965 bytes, last modified: n/a - 100% done
Saving cs-training.csv to cs-training.csv

```
import io
import pandas as pd

df = pd.read_csv(io.BytesIO(uploaded['cs-training.csv']))
df.head() # Shows first 5 rows
```



Unnamed:
0

SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines age NumberOfTir

0	1	2	3	4
0	1	1	0.766127	45
1	2	0	0.957151	40
2	3	0	0.658180	38
3	4	0	0.233810	30

✓	✓	✓	0.233810	30
4	5	0	0.907239	49

```
from google.colab import files
uploaded = files.upload()
```



cs-training.csv

cs-training.csv(application/vnd.ms-excel) - 7564965 bytes, last modified: n/a - 100% done
Saving cs-training.csv to cs-training (1).csv

Start coding or [generate](#) with AI.

```
import io
import pandas as pd

df = pd.read_csv(io.BytesIO(uploaded['cs-training (1).csv']))
df.head() # This shows the first few rows of data
```



	Unnamed: 0	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTir
0	1	1	0.766127	45	
1	2	0	0.957151	40	
2	3	0	0.658180	38	
3	4	0	0.233810	30	
4	5	0	0.907239	49	

```
# Just use this instead
# Fill missing values
df.fillna(df.mean(), inplace=True)

# Separate features and target
X = df.drop('SeriousDlqin2yrs', axis=1)
y = df['SeriousDlqin2yrs']

# Train-test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale features
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
from sklearn.linear_model import LogisticRegression
```

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

▼ LogisticRegression ⓘ ?

LogisticRegression()

```
from sklearn.metrics import classification_report, confusion_matrix
```

```
y_pred = model.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[27975   69]
 [ 1880   76]]
```

		precision	recall	f1-score	support
	0	0.94	1.00	0.97	28044
	1	0.52	0.04	0.07	1956
accuracy				0.94	30000
macro avg		0.73	0.52	0.52	30000
weighted avg		0.91	0.94	0.91	30000

```
uploaded.keys()
```

```
dict_keys(['cs-training (1).csv'])
```

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

