

main

May 29, 2025

```
[1]: #MERGING THE PICKLE FILES
import pandas as pd
import os

folderpath = "/home/ajay/Documents/sleeping_dog_don/FRAUD-DETECTION/dataset/
↳data"
pkl_files = [f for f in os.listdir(folderpath) if f.endswith(".pkl")]
ds = pd.concat([pd.read_pickle(os.path.join(folderpath,files))for files in
↳pkl_files],ignore_index=True)
print(ds.head())
ds.count()
```

	TRANSACTION_ID	TX_DATETIME	CUSTOMER_ID	TERMINAL_ID	TX_AMOUNT	\
0	0	2018-04-01 00:00:31	596	3156	57.16	
1	1	2018-04-01 00:02:10	4961	3412	81.51	
2	2	2018-04-01 00:07:56	2	1365	146.00	
3	3	2018-04-01 00:09:29	4128	8737	64.49	
4	4	2018-04-01 00:10:34	927	9906	50.99	

	TX_TIME_SECONDS	TX_TIME_DAYS	TX_FRAUD	TX_FRAUD_SCENARIO
0	31	0	0	0
1	130	0	0	0
2	476	0	0	0
3	569	0	0	0
4	634	0	0	0

```
[1]: TRANSACTION_ID      1754155
TX_DATETIME              1754155
CUSTOMER_ID              1754155
TERMINAL_ID              1754155
TX_AMOUNT                1754155
TX_TIME_SECONDS          1754155
TX_TIME_DAYS             1754155
TX_FRAUD                 1754155
TX_FRAUD_SCENARIO        1754155
dtype: int64
```

```
[2]: #THE TRANSACTION ID AND DATE TIME ARE NOT DROPPED FOR NOW
```

```
ds = ds.drop(columns=['TRANSACTION_ID','TX_DATETIME'])
```

```
from sklearn.preprocessing import LabelEncoder
```

```
#THE GIVEN DATAFRAME IS CLEAN BUT DOING THIS JUST INCASE
```

```
ds['TERMINAL_ID']= LabelEncoder().fit_transform(ds['TERMINAL_ID'])
```

```
[3]: from sklearn.model_selection import train_test_split
```

```
X = ds.drop(columns=['TX_FRAUD'])
```

```
y = ds['TX_FRAUD']
```

```
X_train , X_test , y_train , y_test = train_test_split(X,y ,test_size=0.  
↪2,random_state=42)
```

```
[4]: from sklearn.ensemble import RandomForestClassifier
```

```
randomforest = RandomForestClassifier(n_estimators=100,random_state=42)
```

```
randomforest.fit(X_train,y_train)
```

```
[4]: RandomForestClassifier(random_state=42)
```

```
[8]: from sklearn.metrics import classification_report ,confusion_matrix
```

```
y_pred = randomforest.predict(X_test)
```

```
print(confusion_matrix(y_test,y_pred))
```

```
print(classification_report(y_test,y_pred))
```

```
[[347970      0]
 [      0 2861]]
```

		precision	recall	f1-score	support
	0	1.00	1.00	1.00	347970
	1	1.00	1.00	1.00	2861
accuracy				1.00	350831
macro avg		1.00	1.00	1.00	350831
weighted avg		1.00	1.00	1.00	350831