Anomaly Detection using the Local Outlier Factor

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn.neighbors import LocalOutlierFactor
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

```
df = pd.read_csv("/content/creditcard.csv")
```

```
df.head()
```

	Time	V1	V2	V3	V4	V5	V6	\
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.23959
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.0788(
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

```
df.describe()
```

	Time	V1	V2	V3	V4
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+0
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-1
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+0(
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+0(
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-0
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-0
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-0
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+0

```
print(df.shape)
data = df.sample(frac = 0.25, random_state = 1) #simple random sampling code
print(data.shape)
```

```
(284807, 31)
```

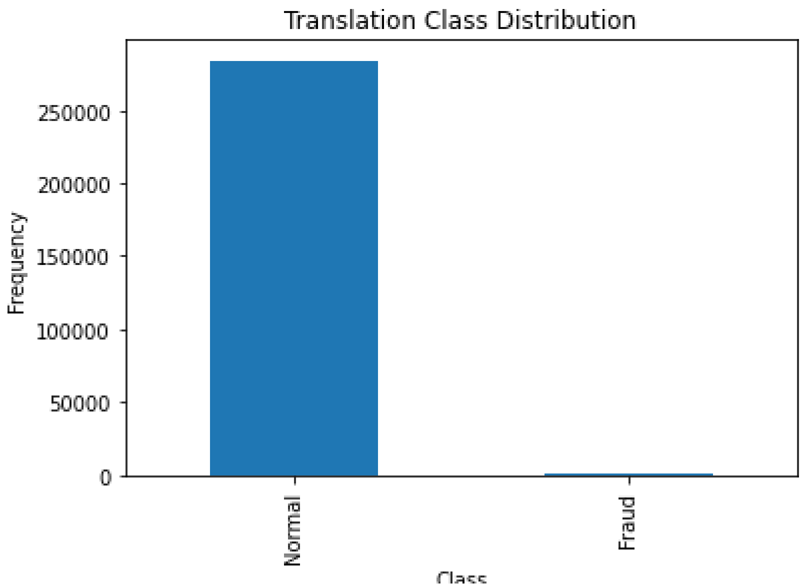
```
(71202, 31)
```

```
data.describe()
```

	Time	V1	V2	V3	V4
count	71202.000000	71202.000000	71202.000000	71202.000000	71202.000000
mean	94589.355636	0.008754	-0.003021	0.002328	0.001971
std	47542.490580	1.950008	1.652787	1.501228	1.414376
min	0.000000	-46.855047	-63.344698	-31.813586	-5.266509
25%	53898.250000	-0.915461	-0.606529	-0.891045	-0.846420
50%	84566.000000	0.034567	0.062170	0.178094	-0.016719
75%	139202.750000	1.319759	0.800553	1.029088	0.745944
max	172784.000000	2.430507	21.467203	4.069865	16.715537

```
num_classes = pd.value_counts(df['Class'], sort = True)
num_classes.plot(kind = 'bar')
plt.title("Translation Class Distribution")
plt.xticks(range(2), ["Normal","Fraud"])
plt.xlabel("Class")
plt.ylabel("Frequency")
```

```
Text(0, 0.5, 'Frequency')
```



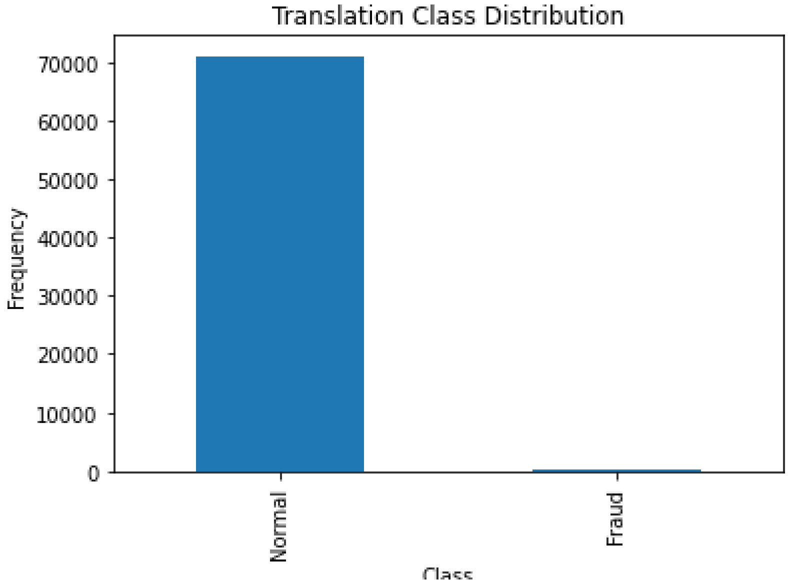

```
fraud = df[df['Class']==1]
normal = df[df['Class']==0]
print(fraud.shape, normal.shape)
```

```
(492, 31) (284315, 31)
```

```
num_classes = pd.value_counts(data['Class'], sort = True)
num_classes.plot(kind = 'bar')
```

```
num_classes.plot(kind = "bar")
plt.title("Translation Class Distribution")
plt.xticks(range(2), ["Normal","Fraud"])
plt.xlabel("Class")
plt.ylabel("Frequency")
```

```
Text(0, 0.5, 'Frequency')
```



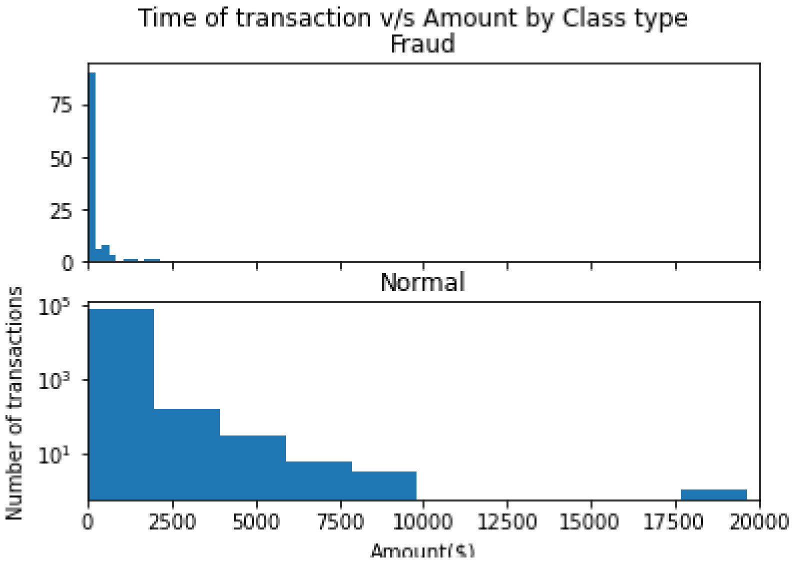

```
fraud_frac = data[data['Class']==1]
normal_frac = data[data['Class']==0]
print(fraud_frac.shape, normal_frac.shape)
```

```
(111, 31) (71091, 31)
```

```
f, (ax1,ax2) = plt.subplots(2,1, sharex = True)
f.suptitle('Time of transaction v/s Amount by Class type')
ax1.scatter(fraud_frac.Time, fraud_frac.Amount)
ax1.set_title('Fraud')
plt.ylabel('Amount')

ax2.scatter(normal_frac.Time,normal_frac.Amount)
ax2.set_title('Normal')
plt.xlabel('Time in secs')
plt.ylabel('Amount')
plt.xlim((0,20000))
```

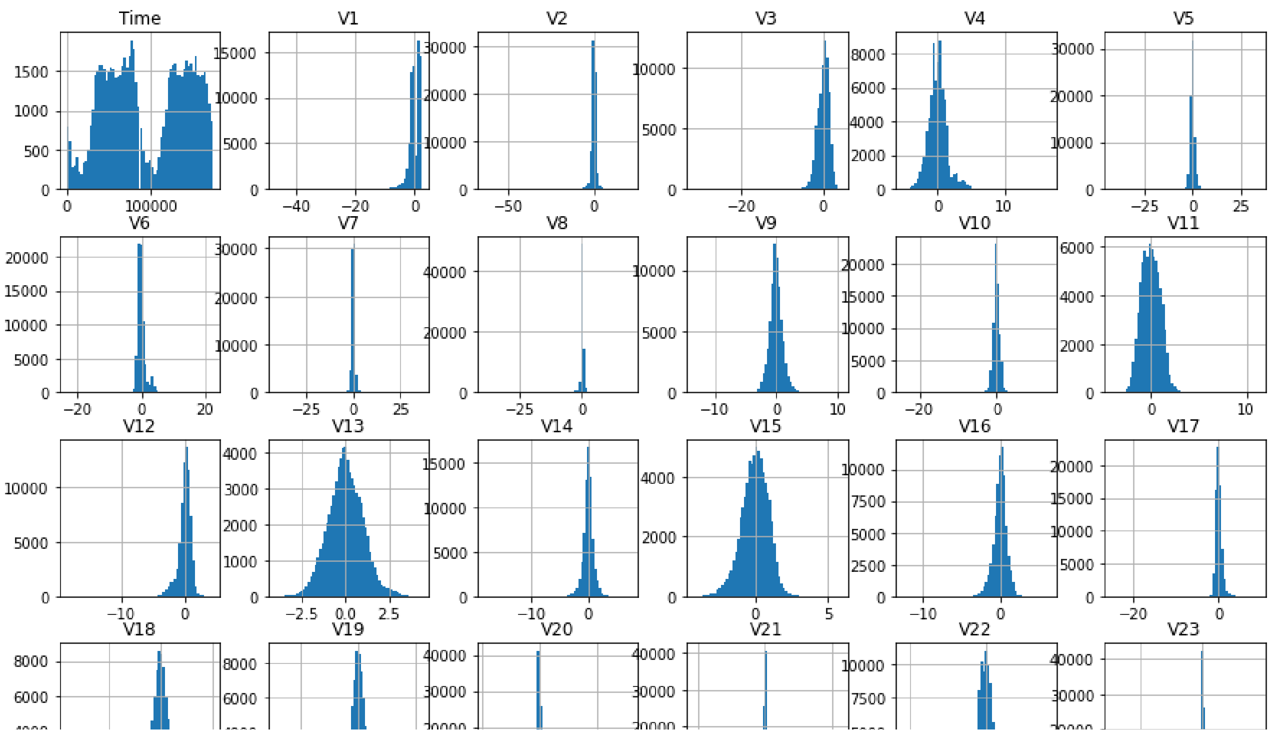
```
(0.0, 20000.0)
```


```
f, (ax1,ax2) = plt.subplots(2,1, sharex = True)
f.suptitle('Time of transaction v/s Amount by Class type')
ax1.hist(fraud_frac.Amount, bins=10)
ax1.set_title('Fraud')
plt.ylabel('Amount')


ax2.hist(normal_frac.Amount, bins=10)
ax2.set_title('Normal')
plt.xlabel('Amount($)')
plt.ylabel('Number of transactions')
plt.xlim((0,20000))
plt.yscale('log')
```




```
#features should be normally distributed for the unsupervised anomally detection algorithm
data.hist(figsize=(15,15), bins=64)
#plotting histograms to check if each feature is normally distributed or not
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28a4b1d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28a70ed0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28a5f710>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f29494e90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f29924e90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f294a3210>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1f292e68d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f292dfe90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f292dfc10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f292b0550>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28e60250>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f29279190>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28e8f610>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f292bcbd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28adf4d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28b70110>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28acfd50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28e89290>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28e0ea10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28d2af90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28b6f650>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28df2cd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f288ad390>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28863a10>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1f288270d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f287dc750>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28791dd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f2874dbd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f2870e190>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28745710>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f1f286fac90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f286be250>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28673bd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f28639190>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f285ef410>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f285a4950>]],
      dtype=object)
```



```
#Divide the data into train and test and then gonna apply the local outlier factor
columns = data.columns.tolist()
target = columns[-1]
features = columns[:-1]
```

```
data.shape
```

```
(71202, 31)
```

```
0.8*71202
```

```
56961.600000000006
```

```
x_train = data.iloc[:56962,1:-1]
x_test = data.iloc[56962:,1:-1]
y_train = data.iloc[:56962,-1]
y_test = data.iloc[56962:,-1]
```

```
#create the model
anomaly_fraction = len(fraud_frac)/len(normal_frac)
model = LocalOutlierFactor(contamination= anomaly_fraction) # will identifier inlier and o
```

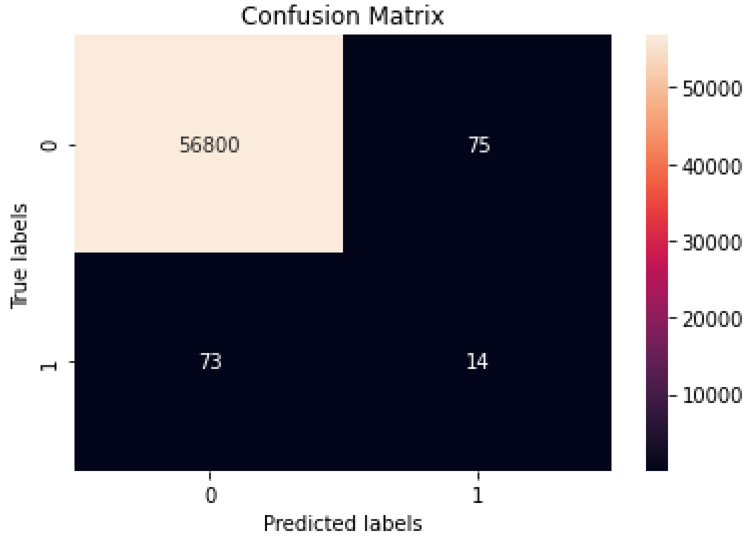
```
#fit the model
y_train_pred = model.fit_predict(x_train)
```

```
#evaluate the model
y_train_pred[y_train_pred ==1] = 0 #(1 changed to 0 as normal in our dataset)
y_train_pred[y_train_pred ==-1] = 1 #(-1 changed to 1 as fraud in out dataset)

y_test_pred = model.fit_predict(x_test)

y_test_pred[y_test_pred ==1]=0
y_test_pred[y_test_pred ==-1]=1
```

```
#plotting 2 confusion matrix, 1 for training and 1 for testing
import seaborn as sns
cm_train = confusion_matrix(y_train, y_train_pred)
ax = plt.subplot()
sns.heatmap(cm_train, annot =True, fmt = 'g', ax=ax)
ax = plt.subplot()
```

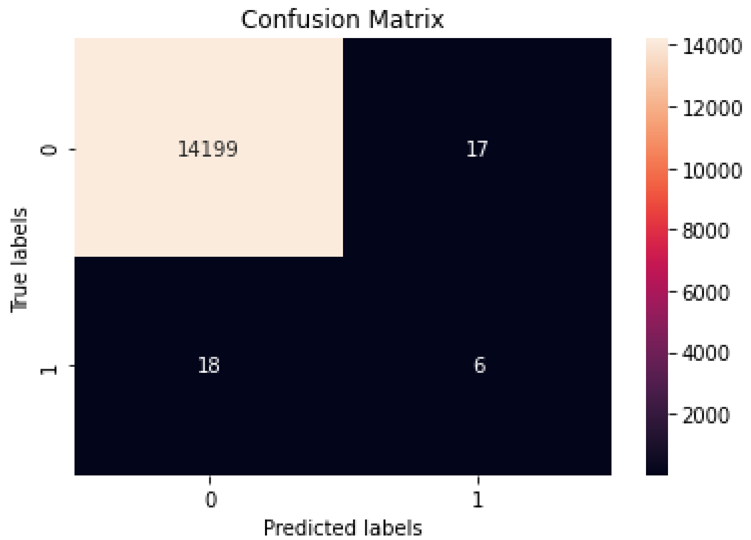
```
ax.set_xlabel('Predicted labels')
ax.set_ylabel('True labels')
ax.set_title('Confusion Matrix')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: MatplotlibDeprecatio

Text(0.5, 1.0, 'Confusion Matrix')
```




```
cm_test = confusion_matrix(y_test, y_test_pred)
ax = plt.subplot()
sns.heatmap(cm_test, annot =True, fmt = 'g', ax=ax)
ax = plt.subplot()
ax.set_xlabel('Predicted labels')
ax.set_ylabel('True labels')
ax.set_title('Confusion Matrix')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: MatplotlibDep
  after removing the cwd from sys.path.
Text(0.5, 1.0, 'Confusion Matrix')
```



out 111 only 14 correctly predicted