databrickscreditcard

Overview

This notebook will show you how to create and query a table or DataFrame that you uploaded to DBFS. DBFS (https://docs.databricks.com/user-guide/dbfs-databricks-file-system.html) is a Databricks File System that allows you to store data for querying inside of Databricks. This notebook assumes that you have a file already inside of DBFS that you would like to read from.

This notebook is written in **Python** so the default cell type is Python. However, you can use different languages by using the `%LANGUAGE` syntax. Python, Scala, SQL, and R are all supported.

```
# File location and type
file_location = "/FileStore/tables/creditcard.csv"
file_type = "csv"

# CSV options
infer_schema = "true"
first_row_is_header = "true"
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
df = spark.read.format(file_type) \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)

display(df)
```

	Time	V1	V2	V3
1	0	-1.3598071336738	-0.0727811733098497	2.53634673796914
2	0	1.19185711131486	0.26615071205963	0.16648011335321
3	1	-1.35835406159823	-1.34016307473609	1.77320934263119
4	1	-0.966271711572087	-0.185226008082898	1.79299333957872
5	2	-1.15823309349523	0.877736754848451	1.548717846511
6	2	-0.425965884412454	0.960523044882985	1.14110934232219
7	4	1.22965763450793	0.141003507049326	0.045370773589944

Truncated results, showing first 1000 rows.

```
# Create a view or table

temp_table_name = "creditcard_csv"

df.createOrReplaceTempView(temp_table_name)
```

```
%sql

/* Query the created temp table in a SQL cell */

select * from `creditcard_csv`
```

	Time	V1	V2	V3
1	0	-1.3598071336738	-0.0727811733098497	2.53634673796914
2	0	1.19185711131486	0.26615071205963	0.16648011335321
3	1	-1.35835406159823	-1.34016307473609	1.77320934263119
4	1	-0.966271711572087	-0.185226008082898	1.79299333957872
5	2	-1.15823309349523	0.877736754848451	1.548717846511
6	2	-0.425965884412454	0.960523044882985	1.14110934232219
7	4	1.22965763450793	0.141003507049326	0.045370773589944

Truncated results, showing first 1000 rows.

```
# With this registered as a temp view, it will only be available to this
particular notebook. If you'd like other users to be able to query this
table, you can also create a table from the DataFrame.
# Once saved, this table will persist across cluster restarts as well as
allow various users across different notebooks to query this data.
# To do so, choose your table name and uncomment the bottom line.

permanent_table_name = "creditcard"
df_fraud=df.select("*").toPandas()

# df.write.format("parquet").saveAsTable(permanent_table_name)
```

```
import os
import mlflow
import logging
import numpy as np
import pandas as pd
import mlflow.pyfunc
import mlflow.sklearn
import seaborn as sns
from sklearn.svm import SVC
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from imblearn.under_sampling import RandomUnderSampler
from sklearn.model_selection import cross_val_score,train_test_split
from hyperopt import
fmin,tpe,hp,SparkTrials,Trials,STATUS_OK,rand,space_eval
```

```
%sh
pip install --upgrade mlflow
pip install -U imbalanced-learn
pip install --upgrade hyperopt
```

```
Requirement already satisfied: mlflow in /databricks/python3/lib/python3.8/s
ite-packages (1.22.0)
Requirement already satisfied: pytz in /databricks/python3/lib/python3.8/sit
e-packages (from mlflow) (2020.5)
Requirement already satisfied: Flask in /databricks/python3/lib/python3.8/si
te-packages (from mlflow) (2.0.2)
Requirement already satisfied: sqlalchemy in /databricks/python3/lib/python
3.8/site-packages (from mlflow) (1.4.27)
Requirement already satisfied: pandas in /databricks/python3/lib/python3.8/s
ite-packages (from mlflow) (1.2.4)
Requirement already satisfied: sqlparse>=0.3.1 in /databricks/python3/lib/py
thon3.8/site-packages (from mlflow) (0.4.2)
Requirement already satisfied: gitpython>=2.1.0 in /databricks/python3/lib/p
ython3.8/site-packages (from mlflow) (3.1.24)
Requirement already satisfied: alembic<=1.4.1 in /databricks/python3/lib/pyt
hon3.8/site-packages (from mlflow) (1.4.1)
Requirement already satisfied: numpy in /databricks/python3/lib/python3.8/si
te-packages (from mlflow) (1.19.2)
Requirement already satisfied: databricks-cli>=0.8.7 in /databricks/python3/
lib/python3.8/site-packages (from mlflow) (0.16.2)
Requirement already satisfied: packaging in /databricks/python3/lib/python3.
```

```
df_fraud.head(5)
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533

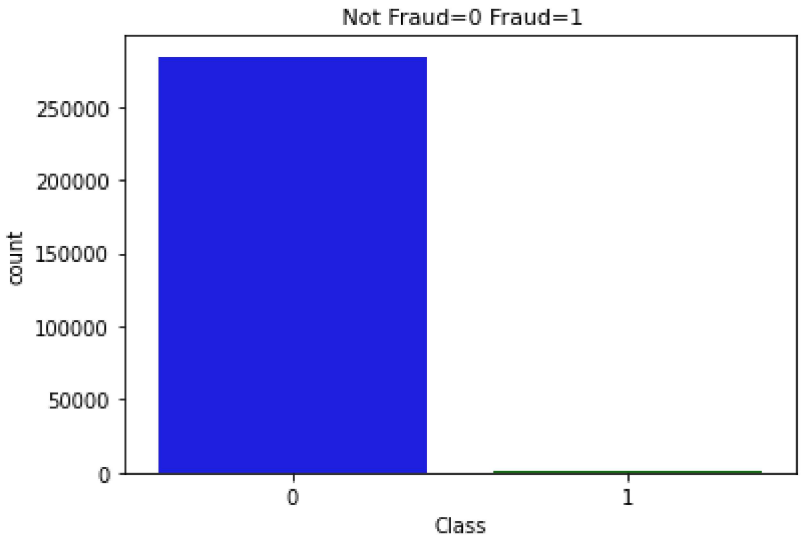
5 rows × 31 columns

```
print("number of rows:"+str(len(df_fraud)))
print("number of columns:"+str(len(df_fraud.columns)))
df_fraud["Class"].unique()
```

```
number of rows:284807
number of columns:31
Out[73]: array([0, 1], dtype=int32)
```

```
colors=["blue","green"]
sns.countplot('Class',data=df_fraud,palette=colors)
plt.title(' Not Fraud=0 Fraud=1',fontsize=11);
```

```
/databricks/python/lib/python3.8/site-packages/seaborn/_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```




```
len(df_fraud[df_fraud["Class"]==0]),len(df_fraud[df_fraud["Class"]==1])
```

```
Out[75]: (284315, 492)
```

```
undersample=RandomUnderSampler(sampling_strategy=0.7)
feature_columns=list(df_fraud.columns.drop('Class'))
X=df_fraud[feature_columns]
Y=df_fraud["Class"]
X_balanced,Y_balanced=undersample.fit_resample(X,Y)
df_fraud=X_balanced
df_fraud["Class"]=Y_balanced


y_balance=pd.DataFrame(Y_balanced,columns=['Class'])
colors=["blue","green"]
sns.countplot('Class',data=y_balance,palette=colors)
plt.title('Not Fraud=0 Fraud=1',fontsize=11)
```

```
/databricks/python/lib/python3.8/site-packages/seaborn/_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
Out[77]: Text(0.5, 1.0, 'Not Fraud=0 Fraud=1')
```


```
len(df_fraud[df_fraud["Class"]==0]),len(df_fraud[df_fraud["Class"]==1])
```

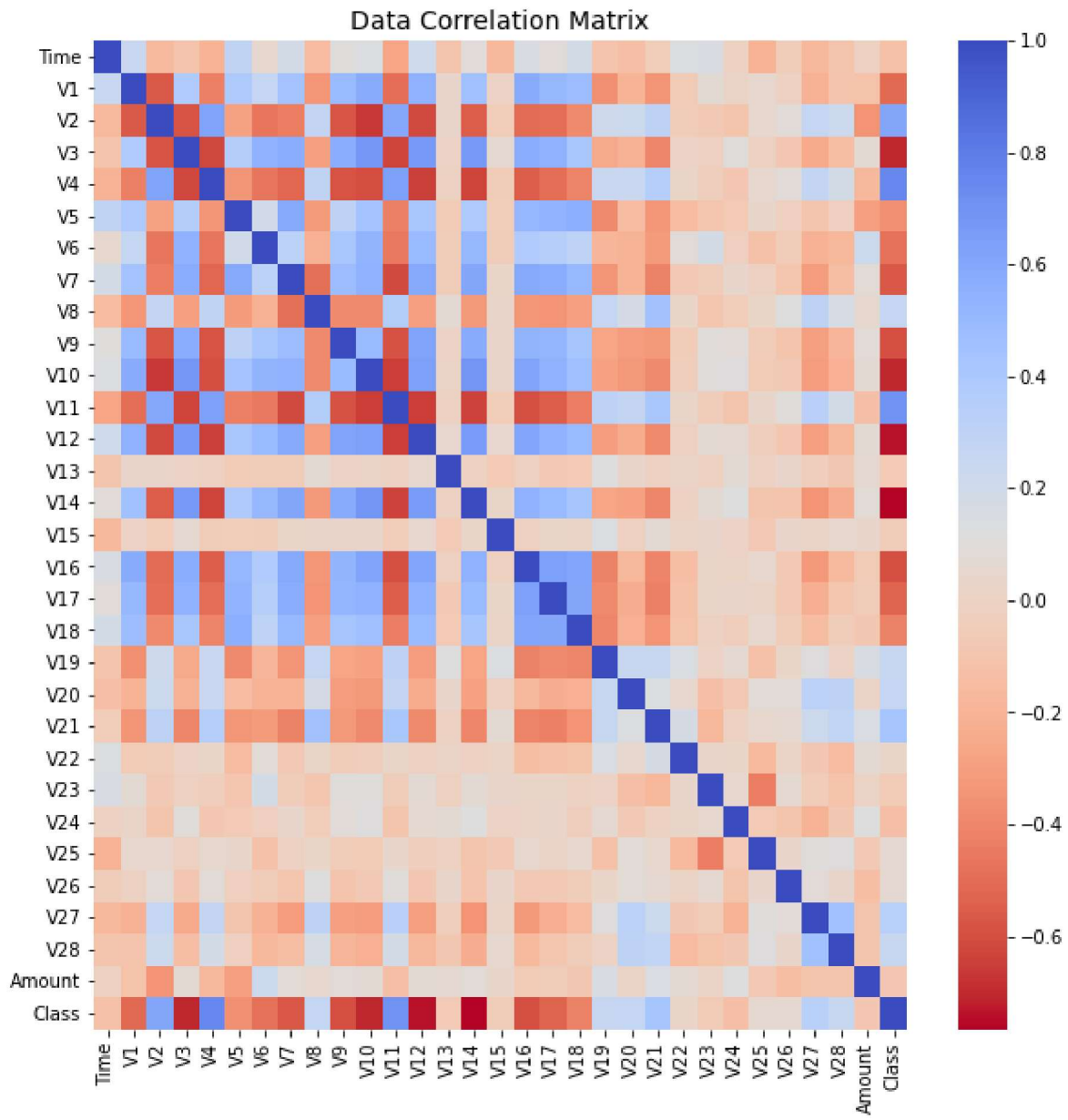
```
Out[78]: (702, 492)
```

```
dims=(4,4)

f, axes=plt.subplots(dims[0],dims[1], figsize(20,20))
axis_i,axis_j=0,0
for col in df_fraud.columns:
    if col == 'Class':
        continue
    sns.boxplot(x=df_fraud["Class"],y=df_fraud[col],ax=axes[axis_i,axis_j])
    axis_j += 1
    if axis_j == dim[1]:
        axis_i += 1
        axis_j = 0
```

```
  NameError: name 'figsize' is not defined
```

```
f,(ax1)=plt.subplots(1,1,figsize=(10,10))
balanced_corr=df_fraud.corr(method='spearman')
sns.heatmap(balanced_corr,cmap='coolwarm_r',annot_kws={'size':12},ax=ax1)
ax1.set_title('Data Correlation Matrix',fontsize=14)
plt.show()
```




```
def data_preprocess(dataframe):
    dataframe=dataframe.dropna()
    dataframe.fillna(dataframe.mean(),inplace=True)
    dataframe=dataframe.reset_index()
    minmax_scaler=MinMaxScaler()
    dataframe['Amount']-=dataframe['Amount'].mean(axis=0)
    dataframe['Amount']/=dataframe['Amount'].std(axis=0)

    return dataframe
```

```
def
get_train_test_data(dataframe,feature_columns,target_column,train_test_split
_ratio,random_speed):
     X=dataframe[feature_columns]
     y=dataframe[target_column]

X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=train_test_spl
it_ratio,random_state=random_speed)

     return X_train,X_test,y_train,y_test


config={
     'feature_columns':feature_columns,
    'target_column':['Class'],
    'random_seed':42,
    'train_test_split_ratio':0.8
}


df=df_fraud
df=data_preprocess(df)
X_train,X_test,y_train,y_test=get_train_test_data(df,config['feature_columns
'],config['target_column'],config['train_test_split_ratio'],config['random_s
eed'])


search_space=hp.choice('model_type',[
    {
        'type':'rf',
        'n_estimators':hp.choice('n_estimators',[100,200]),
    },
    {
        'type':'svm',
        'C':hp.lognormal('SVM_C',0,1.0),
        'kernel':hp.choice('kernel',['linear','rbf'])
    },
    {
        'type':'dtree',
        'max_depth':hp.quniform('max_depth',2,5,1),
        'criterion':hp.choice('criterion',['entropy'])
    }

])
```

```
def train_model(parameters):
    model_type=parameters['type']
    del parameters['type']
    if model_type=='svm':
        model=SVC(**parameters)
        model.fit(X_train,y_train.values.ravel())
    elif model_type == 'dtree':
        model=DecisionTreeClassifier(**parameters)
        model.fit(X_train,y_train.values.ravel())
    elif model_type == 'rf':
        model=RandomForestClassifier(**parameters)
        model.fit(X_train,y_train.values.ravel())
    else:
        return 0
    accuracy =cross_val_score(model,X_train,y_train).mean()
    return {'loss':-accuracy,'status':STATUS_OK,'model_type':model_type}
```

```
token =
dbutils.notebook.entry_point.getDbutils().notebook().getContext().apiToken()
.get()
dbutils.fs.put("file:///root/.databrickscfg","
[DEFAULT]\nhost=https://community.cloud.databricks.com\ntoken =
"+token,overwrite=True)

Wrote 98 bytes.
Out[87]: True
```

```
ERROR:root:RESOURCE_ALREADY_EXISTS: Node named 'db1' already exists
Traceback (most recent call last):
  File "<command-347519352388957>", line 7, in <module>
    experiment_id=mlflow.create_experiment(name=experiment_name)
  File "/databricks/python/lib/python3.8/site-packages/mlflow/tracking/fluen
t.py", line 935, in create_experiment
    return MlflowClient().create_experiment(name, artifact_location, tags)
  File "/databricks/python/lib/python3.8/site-packages/mlflow/tracking/clien
t.py", line 507, in create_experiment
    return self._tracking_client.create_experiment(name, artifact_location,
tags)
  File "/databricks/python/lib/python3.8/site-packages/mlflow/tracking/_trac
king_service/client.py", line 182, in create_experiment
    return self.store.create_experiment(
  File "/databricks/python/lib/python3.8/site-packages/mlflow/store/trackin
g/rest_store.py", line 99, in create_experiment
    response_proto = self._call_endpoint(CreateExperiment, req_body)
  File "/databricks/python/lib/python3.8/site-packages/mlflow/store/trackin
g/rest_store.py", line 56, in _call_endpoint
    return call_endpoint(self.get_host_creds(), endpoint, method, json_body,
response_proto)
```