Twitter Data Analysis

```
import os
import sys
import glob
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import sqlite3
import sqlalchemy
import pickle
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.metrics import precision_recall_fscr
from sklearn.metrics import roc_curve, auc, roc_auc_score
from sklearn.metrics import precision_recall_curve, average_precision_score
from sklearn.metrics import cohen_kappa_score, matthews_corrcoef, log_loss
from sklearn.metrics import mean_squared_log_error, median_absolute_error, explained_variance_score
from sklearn.metrics import max_error, mean_poisson_deviance, mean_gamma_deviance
from sklearn.metrics import mean_tweedie_deviance, d2_tweedie_score, d2_pinball_score, d2_absolute_error_score
%matplotlib inline
%matplotlib notebook
```

1 Processing the Data

Twitter data is stored in JSON format as a flat file. JSON contains user, coordinates, and hashtag information, as well as attributes pertaining to the coordinates, text, and retweets tagged in the tweet. We'll extract some particular features from all of the JSON to further investigate them. We could also store the data in a NoSQL database, but for this project we'll simply read in the JSON file line by line and write to a SQLite database. **Note: I found that using the command line utilities `sqlite3`, `sqlite-utils`, `textutil`, `ogrinfo` (part of `gdal`), and `jq` (`brew install jq`) to be very useful for exploring the data before I wrote any code. Specifically, I found that the `jq` command to flatten the `hashtags` and `urls` arrays and output a new JSON file with only the relevant fields to be much faster than processing the full JSON.** The 2.12 GB file prepared is imported into the database in chunks to avoid memory issues.

```
# Connect to database
conn = sqlalchemy.create_engine('sqlite:///data.db')

# Define chunk size
chunksize = 100000
j = 0
for df in pd.read_table('output.txt', sep='|', header=None, chunksize=chunksize):
    print('Processing chunk {}...'.format(j))
    # Add header
    df.columns = ['id', 'created_at', 'user_id', 'user_name', 'user_screen_name', 'user_location', 'user_description', 'user_verified', 'user_followers_count', 'user_friends_count', 'user_listed_count', 'user_favourites_count', 'user_statuses_count', 'user_created_at', 'text', 'source', 'truncated', 'in_reply_to_status_id', 'in_reply_to_user_id', 'in_reply_to_screen_name', 'geo_coordinates', 'coordinates', 'place_coordinates', 'quoted_status_id', 'is_quote_status', 'retweeted_status_id', 'quoted_status_id.1', 'quote_count', 'reply_count', 'retweet_count', 'favorite_count', 'hashtags', 'urls', 'user_mentions', 'media', 'lang']
    # Write to database
    df.to_sql('all_data', conn, if_exists='append')
    j += 1
```

```
Processing chunk 0...
Processing chunk 1...
Processing chunk 2...
Processing chunk 3...
Processing chunk 4...
Processing chunk 5...
Processing chunk 6...
Processing chunk 7...
Processing chunk 8...
Processing chunk 9...
Processing chunk 10...
Processing chunk 11...
Processing chunk 12...
Processing chunk
```

```
df_de_tweets_df.dropna(subset=['date'])
df_de_tweets_df['date'] = df_de_tweets_df['date'].dt.date
df_de_tweets_df['date'].value_counts().sort_index().plot(figure=fig, title='Tweets in LA')
```

4.1.4 Analysis

Based on the graph above, we can see specific peak tweet days in Los Dic! Dic all these countries.

- **Turkey - 2021-08-02**
- **Germany - 2021-08-02**
- **UK - 2021-08-02**
- **France - 2021-08-02**

On these peak days, we'll dig deeper into the text data for these three countries.

4.2.2 Word Cloud LA

```
stop_words = set(list(STOPWORDS) + ['RT', 'https', 'amp'])
wordcloud_tweets(df_de_tweets_df, stop_words)
```

one
time
thank
day
game
people
today
a
look
love
see
well
new
good
time
thank
you
great
people
year
watch
video
lol
make
feel
first
ever
reminder
good
morning
good
night
good
luck
good
bye
good
job
good
boy
good
girl