Final Project

Tensorflow for Algorithmic Trading
News Feed Sentiment Analysis for Stock Prediction

Bhandarkar, Karan

CSCI E-63 Big Data Analytics

Harvard University Extension School

Prof. Zoran B. Djordjević

INTRODUCTION

- One of the hottest areas in finance these days is Algorithmic Trading, which uses artificial intelligence to sift through massive troves of data to identify signals that humans can’t see.
- Algorithmic trading (automated trading, black-box trading, or simply algo-trading) is the process of using computers programmed to follow a defined set of instructions for placing a trade in order to generate profits at a speed and frequency that is impossible for a human trader.
- It uses natural-language processing to find keywords like company names, and measures when a story is rising up the media food chain, such as from blogs to newswires, to indicate that it may be important enough to act on.
- The goal of the project is to use sentiment analysis on news data to predict stock price.

AREAS COVERED IN THE PROJECT

- Retrieving data from an API
- Reading JSON files into Python Pandas Data Frame.
- Data analysis techniques to clean-up and prepare malformed data.
- Serializing and De-Serializing Python object structures using Pickle module.
- Python Natural Language Tool Kit (NLTK).
- Neural Network setup and Hyper-parameter tuning.
- Visualization

PROBLEM STATEMENT

- Create a Deep Neural Network that analyzes news feeds for a stock and makes price predictions based on the sentiment.

PROJECT SETUP

- Software(detailed instructions in report):

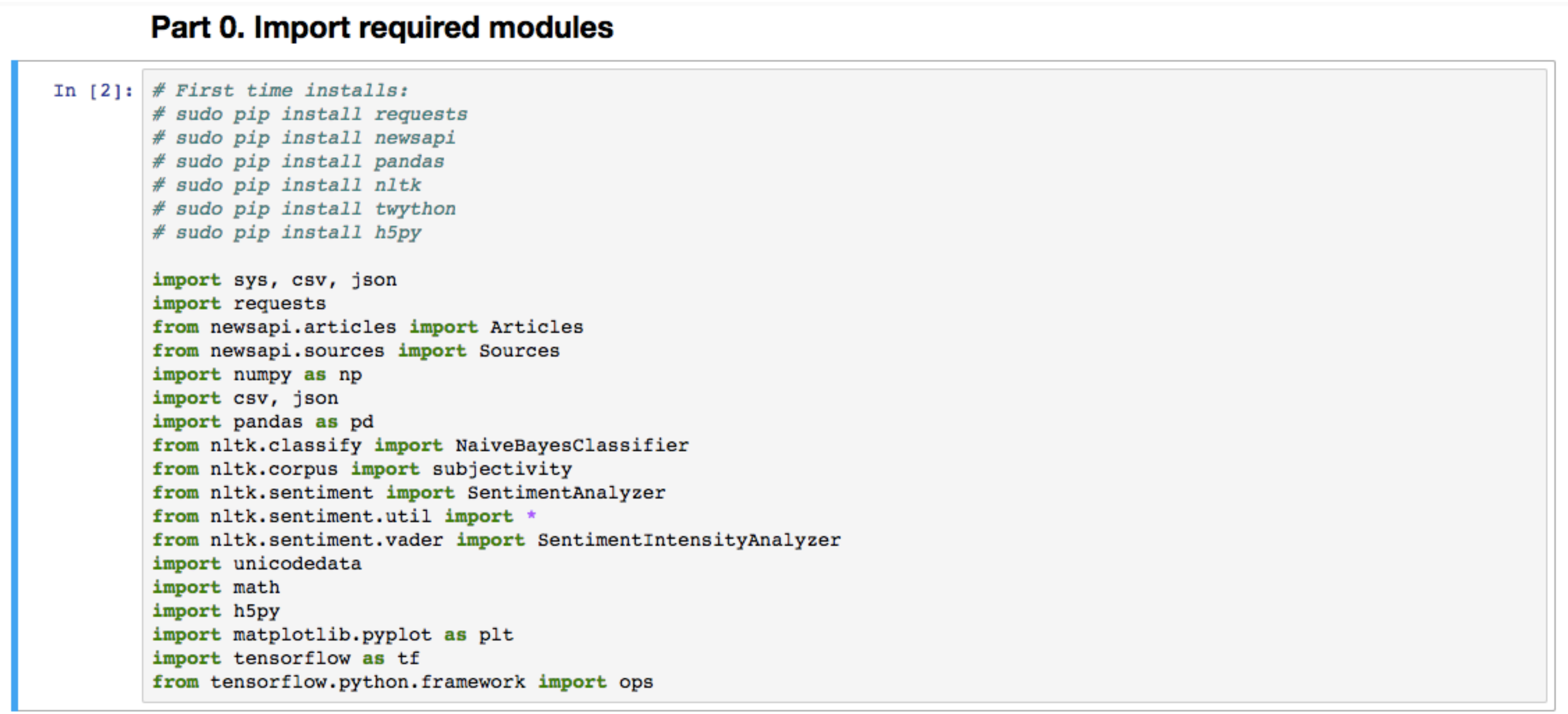
  1. Tensorflow support is available till Python 3.5.2 so a Conda environment with Python 3.5.2 was used.
  2. The project has been built with a Jupyter Notebook running a Python 3 kernel in this environment.
  3. Matplotlib installation is required in this environment for visualizations in the Jupyter Notebook.
  4. Other packages installed: requests, newsapi, pandas, nltk, twython, h5py

- Dataset:

  1. The news articles data set from 2007 to 2016 is downloaded from API calls within the project. Be sure to validate location being dowloaded to and location being used for data retrieval.
  2. The DJIA price list downloaded from the internet is included in the csv format. Update the code to point to downloaded location.

CODE LAYOUT - I

- The code layout is structured and easy to follow.
- The first part is importing the modules used.



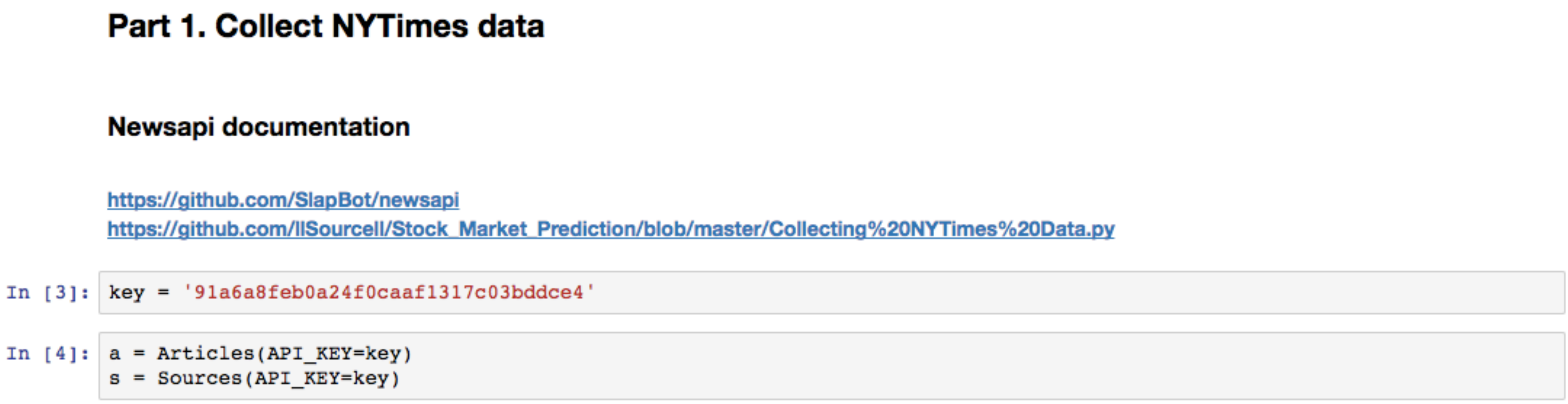
Part 0. Import required modules

In [2]:

```
# First time installs:
# sudo pip install requests
# sudo pip install newsapi
# sudo pip install pandas
# sudo pip install nltk
# sudo pip install twython
# sudo pip install h5py

import sys, csv, json
import requests
from newsapi.articles import Articles
from newsapi.sources import Sources
import numpy as np
import csv, json
import pandas as pd
from nltk.classify import NaiveBayesClassifier
from nltk.corpus import subjectivity
from nltk.sentiment import SentimentAnalyzer
from nltk.sentiment.util import *
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import unicodedata
import math
import h5py
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.python.framework import ops
```

CODE LAYOUT - II

- The second part is collecting the New York Times data from the API from 2007 to 2016.



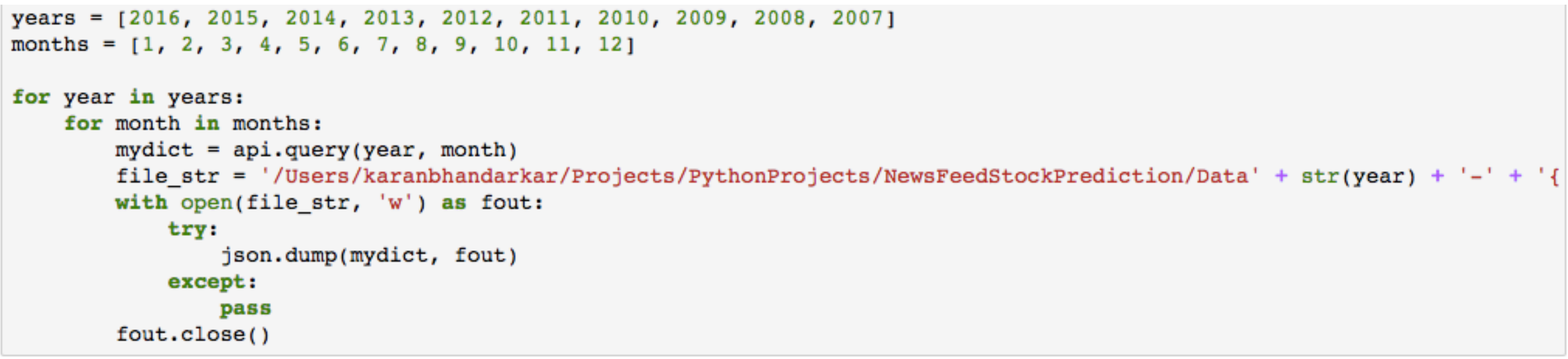
Part 1. Collect NYTimes data

Newsapi documentation

https://github.com/SlapBot/newsapi
https://github.com/llSourcell/Stock_Market_Prediction/blob/master/Collecting%20NYTimes%20Data.py

```
In [3]: key = '91a6a8feb0a24f0caaf1317c03bddce4'
```

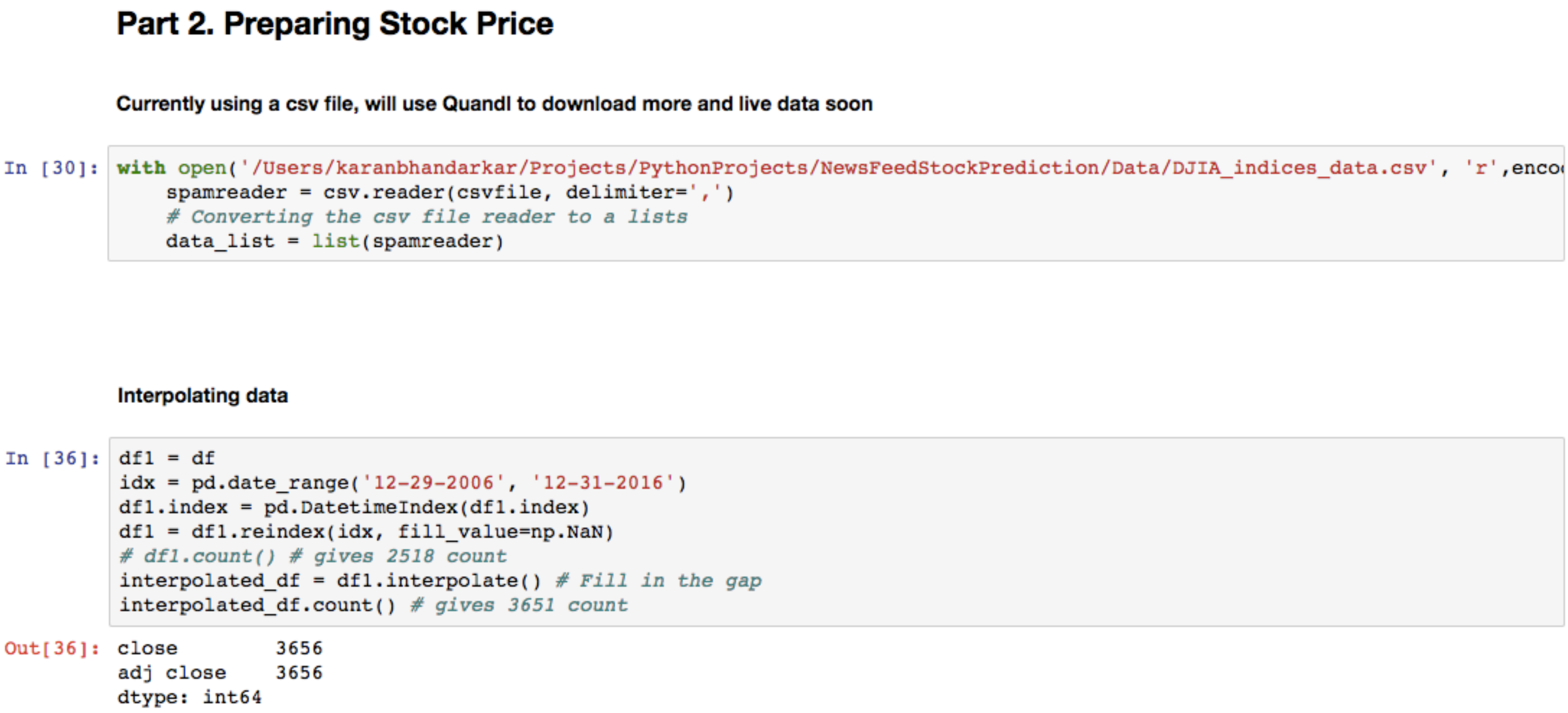
```
In [4]: a = Articles(API_KEY=key)
        s = Sources(API_KEY=key)
```



```
years = [2016, 2015, 2014, 2013, 2012, 2011, 2010, 2009, 2008, 2007]
months = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

for year in years:
    for month in months:
        mydict = api.query(year, month)
        file_str = '/Users/karanbhandarkar/Projects/PythonProjects/NewsFeedStockPrediction/Data' + str(year) + '-' + '{
        with open(file_str, 'w') as fout:
            try:
                json.dump(mydict, fout)
            except:
                pass
        fout.close()
```

CODE LAYOUT - III

- The third part is preparing the DJIA price data for analysis by using interpolation and data selection techniques.



Part 2. Preparing Stock Price

Currently using a csv file, will use Quandl to download more and live data soon

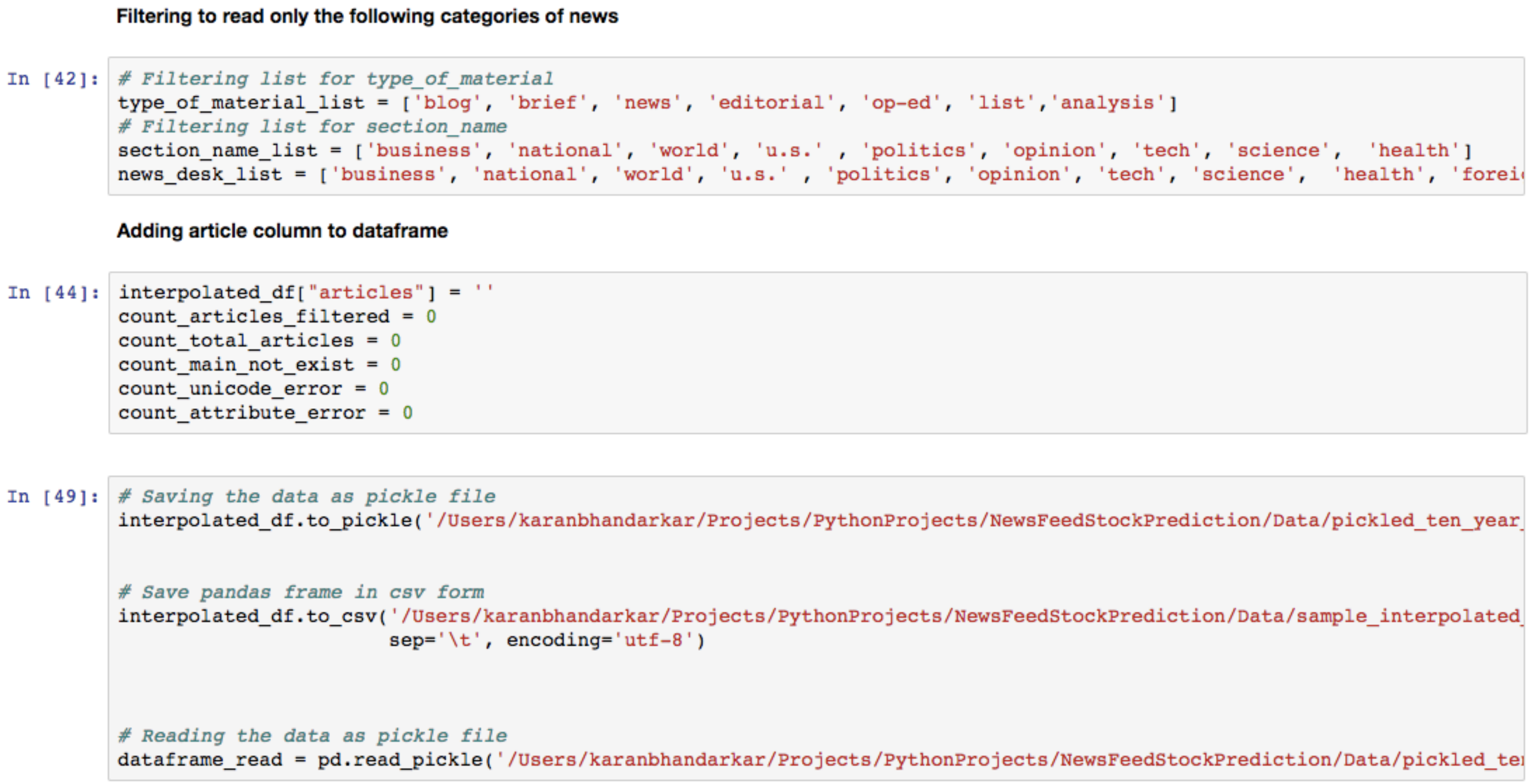
```
In [30]: with open('/Users/karanbhandarkar/Projects/PythonProjects/NewsFeedStockPrediction/Data/DJIA_indices_data.csv', 'r',enco
             spamreader = csv.reader(csvfile, delimiter=',')
             # Converting the csv file reader to a lists
             data_list = list(spamreader)
```

Interpolating data

```
In [36]: df1 = df
         idx = pd.date_range('12-29-2006', '12-31-2016')
         df1.index = pd.DatetimeIndex(df1.index)
         df1 = df1.reindex(idx, fill_value=np.NaN)
         # df1.count() # gives 2518 count
         interpolated_df = df1.interpolate() # Fill in the gap
         interpolated_df.count() # gives 3651 count

Out[36]: close        3656
         adj close    3656
         dtype: int64
```

CODE LAYOUT - IV

- The fourth part is selecting the relevant NYTimes data and merging it all together in a pickle file.
- The pickle module implements a fundamental, but powerful algorithm for serializing and de-serializing a Python object structure.



Filtering to read only the following categories of news

```
In [42]: # Filtering list for type_of_material
         type_of_material_list = ['blog', 'brief', 'news', 'editorial', 'op-ed', 'list','analysis']
         # Filtering list for section_name
         section_name_list = ['business', 'national', 'world', 'u.s.' , 'politics', 'opinion', 'tech', 'science',  'health']
         news_desk_list = ['business', 'national', 'world', 'u.s.' , 'politics', 'opinion', 'tech', 'science',  'health', 'forei
```

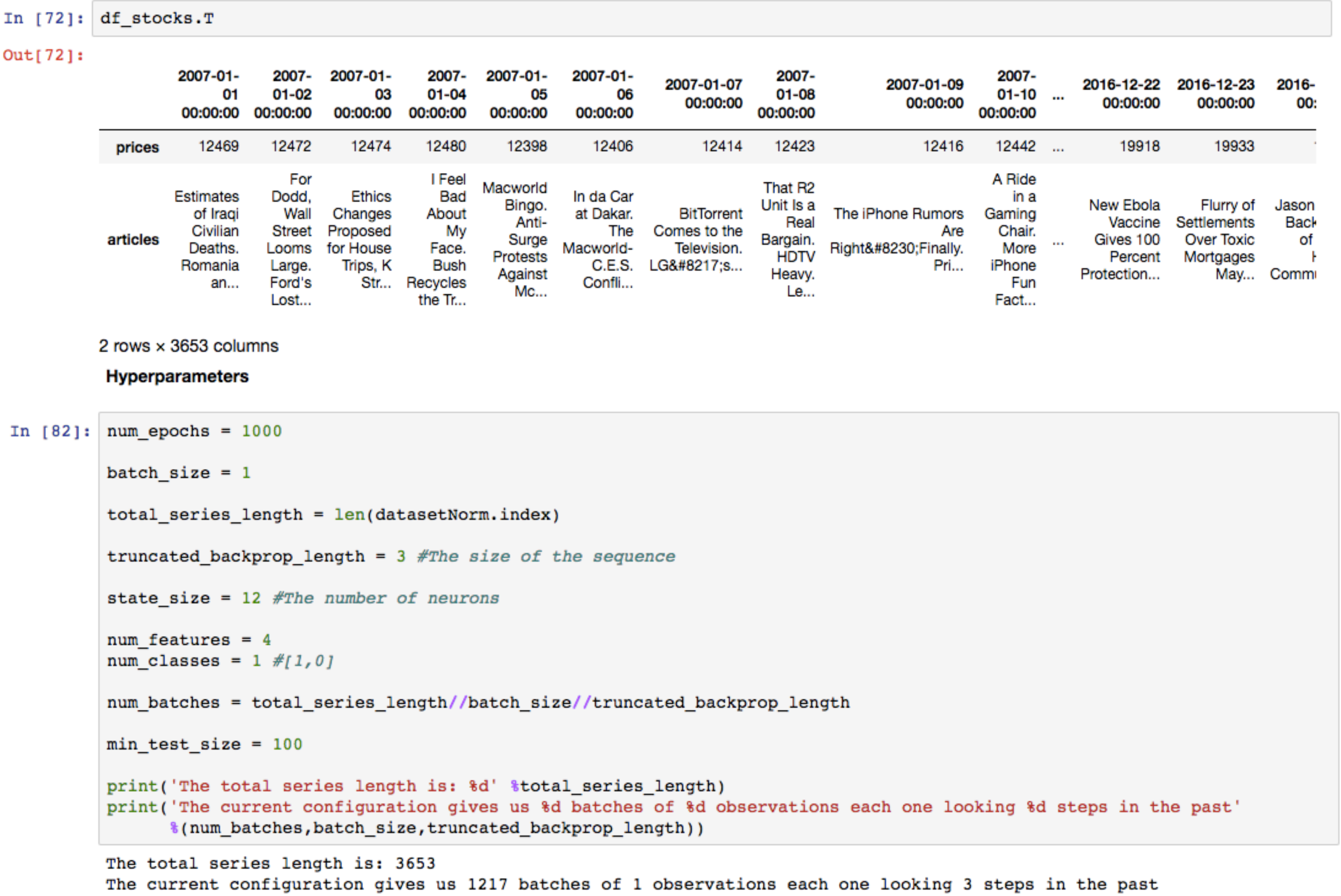
Adding article column to dataframe

```
In [44]: interpolated_df["articles"] = ''
         count_articles_filtered = 0
         count_total_articles = 0
         count_main_not_exist = 0
         count_unicode_error = 0
         count_attribute_error = 0
```

```
In [49]: # Saving the data as pickle file
         interpolated_df.to_pickle('/Users/karanbhandarkar/Projects/PythonProjects/NewsFeedStockPrediction/Data/pickled_ten_year_


         # Save pandas frame in csv form
         interpolated_df.to_csv('/Users/karanbhandarkar/Projects/PythonProjects/NewsFeedStockPrediction/Data/sample_interpolated_
                                sep='\t', encoding='utf-8')



         # Reading the data as pickle file
         dataframe_read = pd.read_pickle('/Users/karanbhandarkar/Projects/PythonProjects/NewsFeedStockPrediction/Data/pickled_te
```

CODE LAYOUT - V

- The final part, the fun part, is the Deep Neural Network trained and then used for prediction. This is where the NLTK package and Tensorflow is used.



```
In [72]: df_stocks.T
```

Out[72]:

	2007-01-01 00:00:00	2007-01-02 00:00:00	2007-01-03 00:00:00	2007-01-04 00:00:00	2007-01-05 00:00:00	2007-01-06 00:00:00	2007-01-07 00:00:00	2007-01-08 00:00:00	2007-01-09 00:00:00	2007-01-10 00:00:00	...	2016-12-22 00:00:00	2016-12-23 00:00:00	2016- 00:
prices	12469	12472	12474	12480	12398	12406	12414	12423	12416	12442	...	19918	19933	
articles	Estimates of Iraqi Civilian Deaths. Romania an...	For Dodd, Wall Street Looms Large. Ford's Lost...	Ethics Changes Proposed for House Trips, K Str...	I Feel Bad About My Face. Bush Recycles the Tr...	Macworld Bingo. Anti-Surge Protests Against Mc...	In da Car at Dakar. The Macworld-C.E.S. Confli...	BitTorrent Comes to the Television. LG&#8217;s...	That R2 Unit Is a Real Bargain. HDTV Heavy. Le...	The iPhone Rumors Are Right&#8230;Finally. Pri...	A Ride in a Gaming Chair. More iPhone Fun Fact...	...	New Ebola Vaccine Gives 100 Percent Protection...	Flurry of Settlements Over Toxic Mortgages May...	Jason Back of Comm

2 rows × 3653 columns

Hyperparameters

```
In [82]: num_epochs = 1000

batch_size = 1

total_series_length = len(datasetNorm.index)

truncated_backprop_length = 3 #The size of the sequence

state_size = 12 #The number of neurons

num_features = 4
num_classes = 1 #[1,0]

num_batches = total_series_length//batch_size//truncated_backprop_length

min_test_size = 100

print('The total series length is: %d' %total_series_length)
print('The current configuration gives us %d batches of %d observations each one looking %d steps in the past'
      %(num_batches,batch_size,truncated_backprop_length))
```

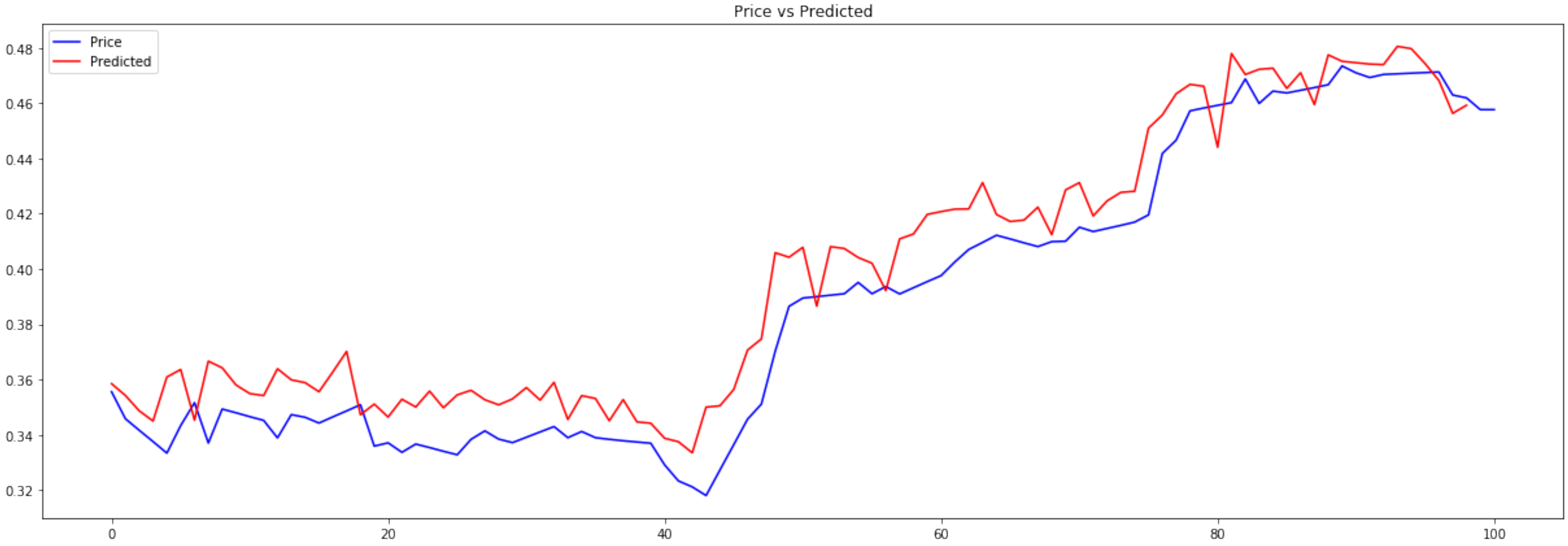
```
The total series length is: 3653
The current configuration gives us 1217 batches of 1 observations each one looking 3 steps in the past
```

CURRENT STATUS

- This project currently runs on static news articles from 2007-2016 made available by New York Times and makes predictions on the price of the Dow Jones Industrial Average (DJIA).
- The Jupyter Notebook downloads the data sets for you using an API call.
- The DJIA open and close prices are used from a csv provided.

RESULTS ACHIEVED

- The predicted prices follow the known prices very closely.
- With a little more optimization in the network architecture, including using different optimizers, loss functions or learning rates, it seems possible to tune this model to predict with trading-worthy accuracy.



FUTURE WORK

- This project currently runs on static news articles from 2007-2016 made available by New York Times and makes predictions on the price of the Dow Jones Industrial Average (DJI).
- The Jupyter Notebook downloads the data sets for you using an API call.
- The DJI open and close prices are used from a csv provided.
- Future work involves:

1. Integrating a dynamic feed of news articles from multiple sources.
2. Parametrizing the stock field for user to specify.
3. Integrating a dynamic feed of real time price information from Quandl.
4. Hosting the project on an AWS EC2 instance far faster run time.

- Detailed implementation for reference and further experiments:

https://github.com/llSourcell/Stock_Market_Prediction

YOUTUBE URLs, Last Page

- Two minute (short): https://youtu.be/ZtwWGCC7z64
- 15 minutes (long): https://youtu.be/eGu36PADdzc