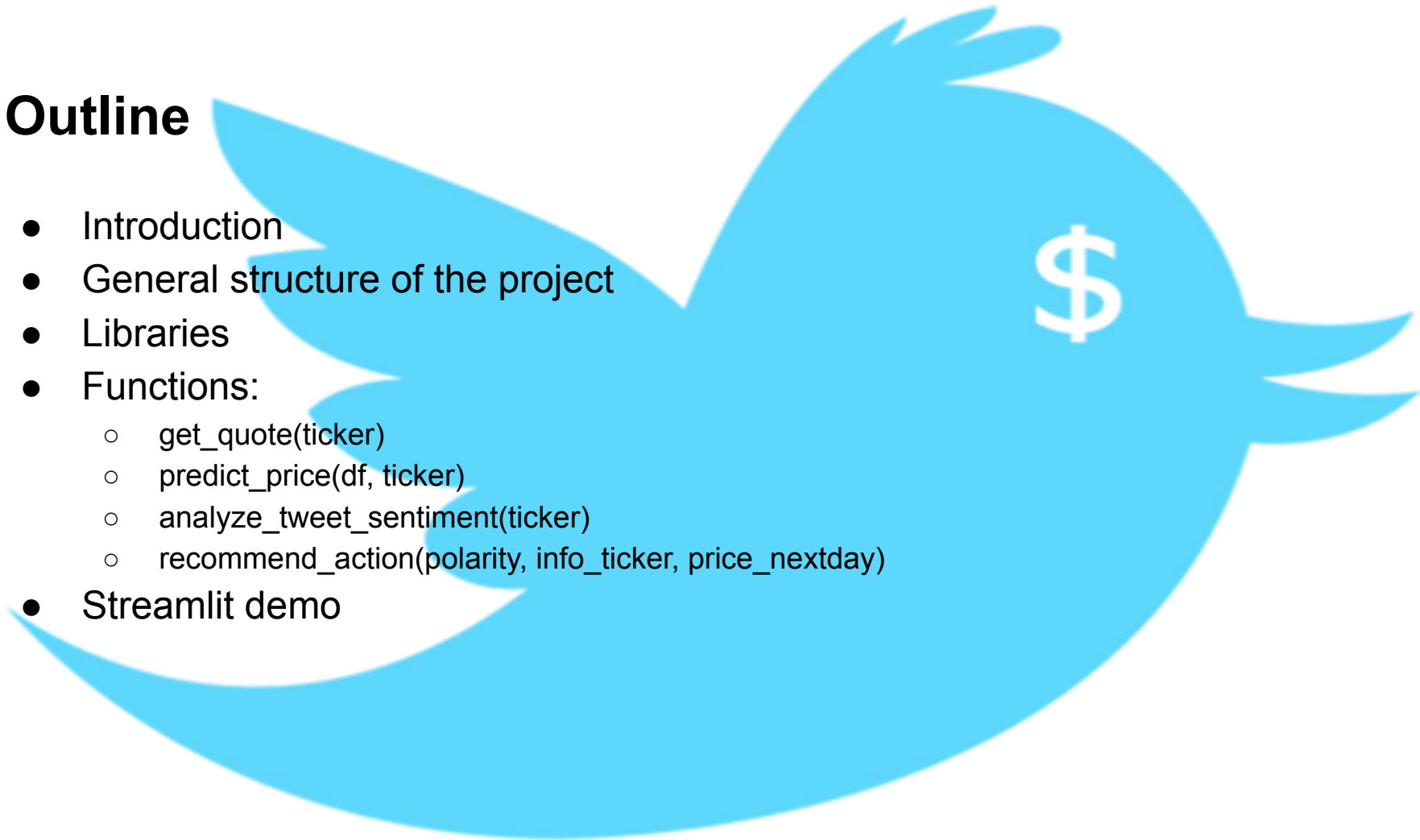


Stock Prediction with Neural Network and Twitter NLP sentiment analysis

CB-DS-06: Quang Vinh Du

Outline

- Introduction
- General structure of the project
- Libraries
- Functions:
 - `get_quote(ticker)`
 - `predict_price(df, ticker)`
 - `analyze_tweet_sentiment(ticker)`
 - `recommend_action(polarity, info_ticker, price_nextday)`
- Streamlit demo



Introduction

- Project inspired by Stock/Index workshop and the NLP twitter group project
- Web tool which will aid in making the decision to buy or sell
- Put into use everything that have been learned in the course:
 - Algorithms
 - Pandas Dataframes
 - Data visualization
 - Neural Network models
 - Web scraping
- **Disclaimer: For educational and information purposes. Does not constitute professional financial advice.**

General structure of the project

1. Import the necessary libraries and Twitter API keys
2. Fetch and store the data for analysis usage
3. Compile and use a neural network to predict the future price of the stock
4. Analyze a number of tweets to get a overall sentiment for the company



Libraries

- **yfinance:** Pythonic way to download market data from Yahoo! Finance's API and intended for personal use only.
- **tweepy:** Open source Python package that gives you a very convenient way to access the Twitter API with Python.
- **textblob:** Python library for processing textual data. Provides common natural language processing (NLP) tasks such as noun phrase extraction, sentiment analysis, classification
- **streamlit:** Open-source app framework for Machine Learning and Data Science teams

Function: get_quote(ticker)

- Input: ticker
String of letters symbolizing a company on the stock market exchange (ie: AAPL, MSFT, TSLA)
- If no CSV file exists for that ticker, it will go fetch the data on Yahoo Finance onto a dataframe and store in a CSV.
- For this project, we will be using data starting 2 years ago.
- Output: info_filename (Path to go get the CSV file for prediction via Neural Network) and ticker_name (Full company's name for that ticker)

Function: predict_price(df, ticker)

- Input: df (Dataframe containing all the ticker's information) and ticker(Symbol)
- Splitting the data into training (80%) and testing sets (20%)
- Model will be using 7 days in the past to try to predict the next day's price
- Very important to scale the training set, convert to np.array for our Recurring Neural Network layer
- Using RNN will allow us to use past outputs as inputs and useful especially in timeseries.
- In particular, we reshaped our array into 3D for the Long short-term memory (LSTM) layer

Function: predict_price(df, ticker)

- Long short-term memory (LSTM):
 - Recurrent neural network (RNN) architecture used in the field of deep learning
 - Can store past important information and forget the information that is not
 - Feedback connections are fitted for sequential data like music, weather, and namely stock market data
- Compile, fit and save the model for future use to save time when checking for a previously searched ticker

Function: predict_price(df, ticker)

- In the same manner as our training set, we need to scale, convert into array and reshape into 3D to test our model
- Plot the real y and the y_{test} onto the test set (100 days)
- Determine the Root Mean Square Error (RMSE):
 - Residuals are a measure of how far from the regression line data points are
 - RMSE is a measure of how spread out these residuals are.
 - It tells us how concentrated the data is around the line of best fit.
 - The lower the value, the better.
- Use the model to predict the next day's price
- Outputs: Next day's price, RMSE, RNN model h5 file path

Function: analyze_tweet_sentiment(ticker)

- Input: ticker (symbol)
- Translate the symbol into full company's name
- Access Twitter API by authenticating with access keys
- Search for n number of tweets and analyze each one of them for polarity:
 - Preprocessing necessary to eliminate irrelevant characters
 - Use Textblob API to determine the polarity of the tweet
 - In the same time populating a list of tweets and its polarity for visualization purposes
 - Polarity < 0 : negative sentiment
 - Polarity = 0 : neutral sentiment
 - Polarity > 0 : positive sentiment
- Counting the number of positive, negative and neutral tweets
- Visualize it via a pie chart and save the image

Function: analyze_tweet_sentiment(ticker)

- Outputs:
 - Overall polarity value from our batch of tweets
 - Polarity verdict to state either its positive or negative overall
 - The number of positive tweets
 - The number of negative tweets
 - The number of neutral_tweets
 - A list of tweets and its polarity values
 - A file path to fetch our pie chart



Streamlit demo

<http://10.0.0.123:8501/>

Stock Prediction with Neural Network and Twitter NLP sentiment analysis

Type in the selected ticker

Search

Loading data...Done!

Today's Apple Inc. (AAPL) information for 2022-04-21

References

- **Machine Learning to Predict Stock Prices:**
<https://towardsdatascience.com/predicting-stock-prices-using-a-keras-lstm-model-4225457f0233>
- **Twitter Sentiment Analysis using Python**
<https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/>
- **Streamlit 101: An in-depth introduction:**
<https://towardsdatascience.com/streamlit-101-an-in-depth-introduction-fc8aad9492f2>