FINANCIAL SENTIMENT ANALYSIS

Matthew Rajan, Nihal Mitta, Jack Orr

INTRODUCTION

As technology is rapidly improving, the Finance world has been eagerly adopting new techniques and algorithms to help get an edge over other players in the market. Generating signals for new trades has become a new research area and we plan to join the effort through the use Machine Learning algorithms for Financial News Sentiment Analysis

DEVELOPMENT PROCESS

Step 1

Get labeled financial statements from Kaggle and Financial PhraseBank

Step 2

Clean data and preprocess it into a bag of words vectorization

Step 3

Implement our algorithms using NumPy and PyTorch

Step 4

Train and test our algorithms and tune hyperparameters

Step 5

Scrape financial news articles and backtest our trading strategy

DATA COLLECTION & PRE-PROCESSING

DATASETS

Financial Sentiment Analysis

- Dataset containing 5322 financial news statements labeling statements as positive, negative, or neutral.
- “Net sales totaled EUR 93.6 mn , up from EUR 93.2 mn in the corresponding period in 2005 “ (Positive)

Financial Phrase Bank v1.0:

- Dataset containing 4840 financial sentences with 5-8 positive, negative, and neutral annotations for each sentence
- Separated into 4 categories:
  - 100% of annotations agree
  - **75% of annotations agree**
  - 66% of annotations agree
  - 50% of annotations agree

DATA COLLECTION/MANIPULATION

- Combined the two datasets
  - Tested all sorts of combinations and found that the Kaggle + 75% Agree works the best
- Cleaned the data
  - Removed non-english words
  - Converted to lowercase
  - Removed numbers, special characters, random symbols, and white spaces

DATA

```
According to Gran , the company has no plans to move all production to Russia , although that is where the company is growing .@neutral
With the new production plant the company would increase its capacity to meet the expected increase in demand and would improve the use of raw materials and therefore increase
For the last quarter of 2010 , Componenta 's net sales doubled to EUR131m from EUR76m for the same period a year earlier , while it moved to a zero pre-tax profit from a pre-t
In the third quarter of 2010 , net sales increased by 5.2 % to EUR 205.5 mn , and operating profit by 34.9 % to EUR 23.5 mn .@positive
Operating profit rose to EUR 13.1 mn from EUR 8.7 mn in the corresponding period in 2007 representing 7.7 % of net sales .@positive
Operating profit totalled EUR 21.1 mn , up from EUR 18.6 mn in 2007 , representing 9.7 % of net sales .@positive
TeliaSonera TLSN said the offer is in line with its strategy to increase its ownership in core business holdings and would strengthen Eesti Telekom 's offering to its customer
STORA ENSO , NORSKE SKOG , M-REAL , UPM-KYMMENE Credit Suisse First Boston ( CFSB ) raised the fair value for shares in four of the largest Nordic forestry groups .@positive
A purchase agreement for 7,200 tons of gasoline with delivery at the Hamina terminal , Finland , was signed with Neste Oil OYj at the average Platts index for this September p
Finnish Talentum reports its operating profit increased to EUR 20.5 mn in 2005 from EUR 9.3 mn in 2004 , and net sales totaled EUR 103.3 mn , up from EUR 96.4 mn .@positive
Clothing retail chain Sepp+ñl+ñ 's sales increased by 8 % to EUR 155.2 mn , and operating profit rose to EUR 31.1 mn from EUR 17.1 mn in 2004 .@positive
Consolidated net sales increased 16 % to reach EUR74 .8 m , while operating profit amounted to EUR0 .9 m compared to a loss of EUR0 .7 m in the prior year period .@positive
```

0	according to the company has no plans to move all production to russia although that is where the company is growing
1	with the new production plant the company would increase its capacity to meet the expected increase in demand and would improve the use of raw materials and therefore increase the production profitability
2	for the last quarter of s net sales doubled to from for the same period a year earlier while it moved to a zero pretax profit from a pretax loss of
3	in the third quarter of net sales increased by to mn and operating profit by to mn
4	operating profit rose to mn from mn in the corresponding period in representing of net sales
5	operating profit totalled mn up from mn in representing of net sales
6	said the offer is in line with its strategy to increase its ownership in core business holdings and would strengthen s offering to its customers
7	credit suisse first boston raised the fair value for shares in four of the largest nordic forestry groups

BAG OF WORDS

- Reads in sentences from dataset and gets the frequency of each word
- Creates a vector of numbers that represents the frequency of each word in the dataset

```
data = pd.read_csv('Training Data/Appended75.csv')
data = data.drop(data.columns[0], axis=1)

# Vectorize the sentences using bag-of-words model
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data['sentence'])
```

MODELS

MODELS

- Logistic Regression
- Multi-Layer Perceptron (MLP)
  - Algorithm
  - Network
- Recurrent Neural Network (RNN)

LOGISTIC REGRESSION

- Linear predictor useful for classification
- Minimizes cross entropy loss
  - Mathematically equivalent to maximizing log-likelihood
  - Iteratively using gradient descent to minimize
- Softmax activation function to calculate probabilities

Helpful Article

```python
import numpy as np


class LogisticRegression:
    def __init__(self, learning_rate=0.01, n_iter=400):
        self.learning_rate = learning_rate
        self.n_iter = n_iter

    def fit(self, X, y):
        self.X = X
        self.y = y
        self.theta = np.zeros((X.shape[1], 3))

        for i in range(self.n_iter):
            z = np.dot(X, self.theta)
            h = self.softmax(z)
            gradient = np.dot(X.T, (h - y)) / len(X)
            self.theta -= self.learning_rate * gradient

    def predict(self, X):
        z = np.dot(X, self.theta)
        h = self.softmax(z)
        predictions = np.argmax(h, axis=1)
        return predictions

    def compute_loss(self, X, y):
        z = np.dot(X, self.theta)
        h = self.softmax(z)
        loss = -np.mean(np.sum(y * np.log(h), axis=1))
        return loss

    def softmax(self, z):
        exp_z = np.exp(z)
        return exp_z / np.sum(exp_z, axis=1, keepdims=True)
```

MLP ALGORITHM

- Utilizes the Multi-Layer Perceptron update rule
  - Predict label with current model weights
  - If prediction is correct do nothing!
  - If wrong:
    - Subtract features of X from weights for predicted class
    - Add features of X from weights for correct class
- Linear predictor

```python
import numpy as np


class MLPNumpy:
    def __init__(self, num_epochs):
        self.num_epochs = num_epochs
        self.num_classes = 3

    def train(self, X, y):
        self.X = X
        self.y = y

        # Add bias
        X = np.concatenate((X, np.ones((X.shape[0], 1))), axis=1)

        # Initialize weight matrix with zeros
        w = np.zeros((self.num_classes, X.shape[1]))

        for epoch in range(self.num_epochs):
            for i in range(X.shape[0]):
                scores = np.dot(w, X[i])

                y_pred = np.argmax(scores)
                y_true = y[i]

                if y_pred != y_true:
                    w[y_true] += X[i]
                    w[y_pred] -= X[i]

        self.w = w

    def predict(self, X):
        self.X = X
        # Add bias
        X = np.concatenate((X, np.ones((X.shape[0], 1))), axis=1)

        scores = np.dot(self.w, X.T)

        y_pred = np.argmax(scores, axis=0)

        return y_pred
```

MLP NETWORK

- "Feed Forward" or "Artificial Neural Network"
- A network of one or more hidden layers of computing nodes
- Each layer is connected by activations
- Great for non-linear classifications

```python
import torch
import torch.nn as nn


class MLP(nn.Module):
    def __init__(self, input_size, hidden_size=256, num_classes=3):
        super(MLP, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.fc2 = nn.Linear(hidden_size, hidden_size)
        self.fc3 = nn.Linear(hidden_size, num_classes)

    def forward(self, x):
        x = x.float()
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

RNN

- Able to handle sequential data
  - Perfect for NLP
- Uses same set of weights across all time-steps
  - The output depends on the current input and the previous hidden states
  - Captures dependencies between inputs

```python
import torch.nn as nn


class RNN(nn.Module):
    def __init__(
            self,
            input_size: int,
            hidden_size: int = 256,
            n_layers: int = 2,
            n_classes: int = 3
    ):
        super().__init__()
        self.n_layers = n_layers
        self.hidden_size = hidden_size
        self.rnn = nn.RNN(input_size, hidden_size, n_layers, batch_first=True)
        self.fc = nn.Linear(hidden_size, n_classes)

    def forward(self, x):
        x = x.float()
        outputs, _ = self.rnn(x)
        outputs = self.fc(outputs)
        return outputs
```

MODEL EVALUATION

MODEL ACCURACY

	Logistic Regression	MLP Algorithm	MLP Network	RNN
Accuracy (%)	82.6789	78.0527	82.3815	81.0884

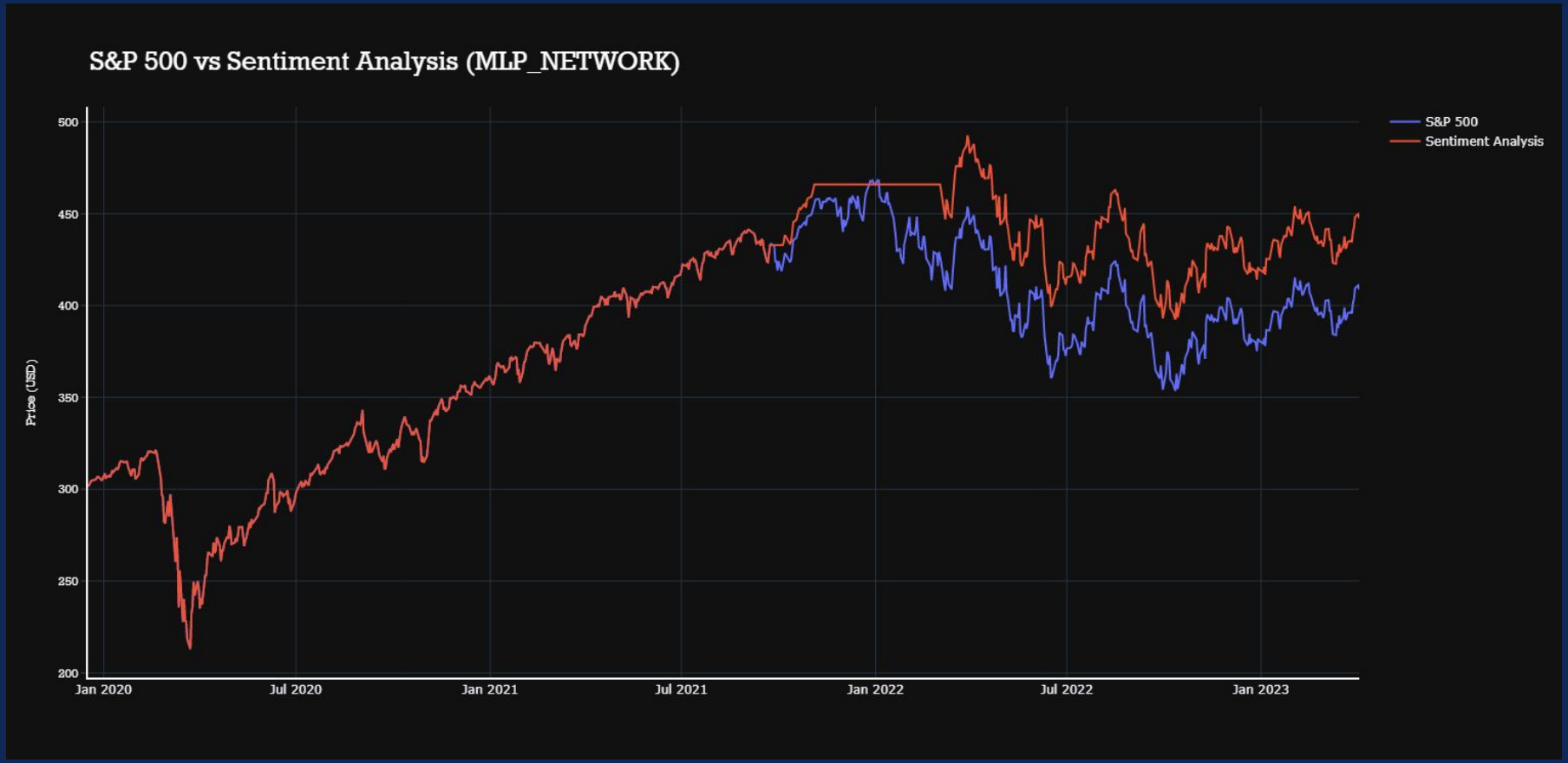
LOGISTIC REGRESSION

MLP Algorithm

MLP Algorithm

	precision	recall	f1-score	support
0	0.48	0.44	0.46	278
1	0.83	0.85	0.84	1052
2	0.82	0.83	0.83	529
accuracy			0.78	1859
macro avg	0.71	0.70	0.71	1859
weighted avg	0.78	0.78	0.78	1859

MLP NETWORK


S&P 500 vs Sentiment Analysis (MLP_NETWORK)
S&P 500
Sentiment Analysis
Price (USD)
500
450
400
350
300
250
200
Jan 2020
Jul 2020
Jan 2021
Jul 2021
Jan 2022
Jul 2022
Jan 2023

RNN


S&P 500 vs Sentiment Analysis (RNN)
S&P 500
Sentiment Analysis
Price (USD)
500
450
400
350
300
250
200
Jan 2020
Jul 2020
Jan 2021
Jul 2021
Jan 2022
Jul 2022
Jan 2023

AREAS TO IMPROVE

- Optimizing and testing various hyperparameters
  - Learning rate, epochs, iterations, etc.
- Playing around with number of layers and types of activation functions
- Trying new models
  - LSTM to improve RNN
- Test new datasets and try cleaning more
  - Mess around with what stopwords to include
- Improving negative sentiment predictions
  - Include more negative examples
- Try different types of encoding
  - Word2Vec, Word embeddings, etc.

THANK YOU

Any Questions?