Stock Price Prediction with Daily News

GU Jinshan 14110914D
MA Mingyu Derek 14110562D
MA Zhenyuan 14111439D
ZHOU Huakang 15050698D

Contents

1. Work flow of the prediction tool
2. Model performance evaluation
3. Robot design and implementation
4. Prediction with Deep learning

Our task

Given:

- 25 news titles for each day, 1900 days
- Each day labeled 1 (stock price increased) or 0 (decreased)

Main Task:

- Given some news titles of a day, what is the change of stock price? (1/0)
- High accuracy of prediction.

Work flow of our prediction tool

Data import

- use python3
- Data import using pandas
- Split data into training set and testing set


```
training_set = data[data['Date'] <= '2014-12-31']
testing_set = data[data['Date'] >= '2015-01-02']
print('done')
```

Data preprocessing

1. Remove all punctuations except hyphens (-)
2. Titles grouping

   i.e. Date 1 Title 1: “AB C” , D1T2: “ED FG” → D1 Title: “AB C ED FG”

3. Remove stop words
4. Stanford corenlp: Replace words with its lemma
5. Vectorization with CountVectorizer(ngram_range=(2,2))
6. Best 2000 Features (word) selection using chi square function
7. We can start to train and test our models now!

Evaluation tools

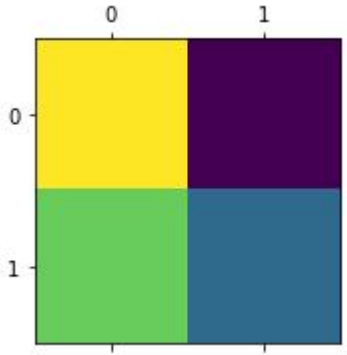
1. Precision, Recall, F1-Score, Support
2. Confusion matrix and heatmap
3. ROC curve and AUC

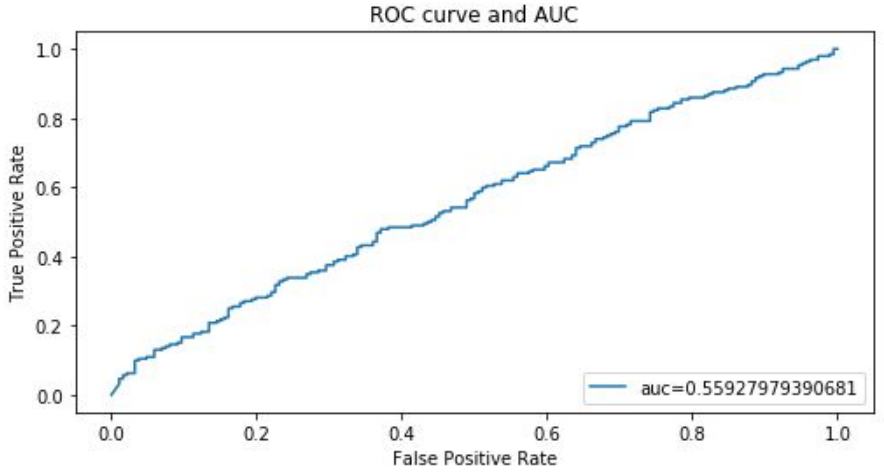
```
classification report:
              precision    recall  f1-score   support

           0       0.44      0.24      0.31       186
           1       0.49      0.71      0.58       192

avg / total       0.47      0.48      0.45       378

Confusion matrix
[[ 44 142]
 [ 55 137]]
```





Models performance evaluation

Models

Try as many as possible!

1. Naive Bayes: Multinomial NB/Bernoulli NB/Gaussian NB
2. Random Forest
3. Support Vector Classification: Linear SVC/Nu-SVC
4. Neural Network Multi-layer Perceptron classifier: LBFGS/SGD/ADAM
5. Decision Tree
6. Gradient Boosting Machines
7. Ada Boost Classifier

Models

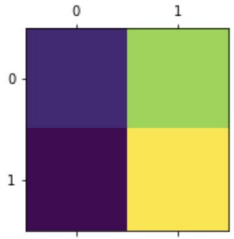
Example Confusion Matrix

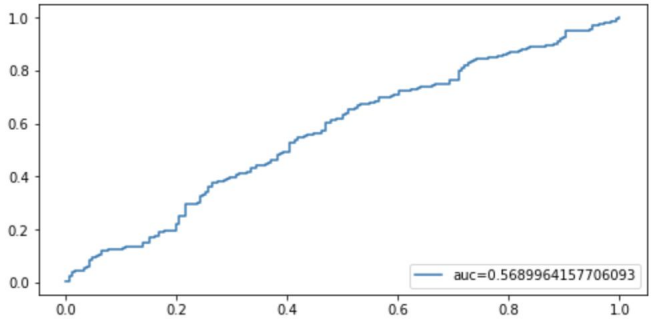
```
MLPClassifier sgd
classification report:
             precision    recall  f1-score   support

          0       0.55      0.30      0.39       186
          1       0.53      0.77      0.63       192

avg / total       0.54      0.54      0.51       378

Confusion matrix
[[ 56 130]
 [ 45 147]]
```





Models

Comparing Precision of All Models

Model	Precision with Corenlp	Precision without Corenlp
Multinomial Naive Bayes	0.47	0.53
Bernoulli Naive Bayes	0.45	0.53
Random Forest	0.50	0.53
Gradient Boosting Machines	0.48	0.46
Ada Boost Classifier	0.55	0.47
Gaussian Naive Bayes	0.49	0.53
Linear SVC	0.49	0.53
Nu-SVC	0.48	0.50
MLPClassifier lbfgs	0.48	0.52
MLPClassifier sgd	0.56	0.54
MLPClassifier adam	0.44	0.52
Decision Tree	0.52	0.51

N-Day Shift analysis

- N-day shift analysis is needed
  - Market may response to the news later
  - Or market may predict the news and act eariler
- Using **previous day news** to predict **today stock** achieves best performance

	no shift	1 day earlier	2 days earlier	1 day later	2 days later
MultinomialNB	0.53	0.51	0.49	0.49	0.48
BernoulliNB	0.53	0.52	0.48	0.5	0.51
RandomForest	0.49	0.52	0.49	0.49	0.5
GradientBoostingClassifier	0.47	0.52	0.52	0.54	0.52
AdaBoostClassifier	0.47	0.52	0.47	0.53	0.48
GaussianNB	0.53	0.54	0.49	0.5	0.53
svm NuSVC	0.5	0.53	0.5	0.5	0.54
MLPClassifier lbfgs	0.52	0.54	0.49	0.48	0.58
MLPClassifier sgd	0.54	0.54	0.51	0.51	0.55
MLPClassifier adam	0.5	0.53	0.51	0.5	0.55
DecisionTree	0.49	0.53	0.52	0.52	0.55
AVG PRECISION	0.50636363	**0.527272727**	0.497272727	0.505454545	0.52636363

Investment Robot

Investment Robot

Assumption:

1. The stock price only increases or decreases for 1% on each day
2. Initial money is $10000

Investment Robot

Two algorithms:

1. ***Buy all or sell all***

   if the prediction result is “increase”, spend all to buy stock;
   otherwise, sell all stocks.

2. ***Cautious approach***

   if the prediction result is “increase” with p probability, put p% money in stock market;
   otherwise keep p% money as cash.

Investment Robot

Experiment:

Simulate the investment for 2 years every time.
Use the data before start time.

Investment Robot

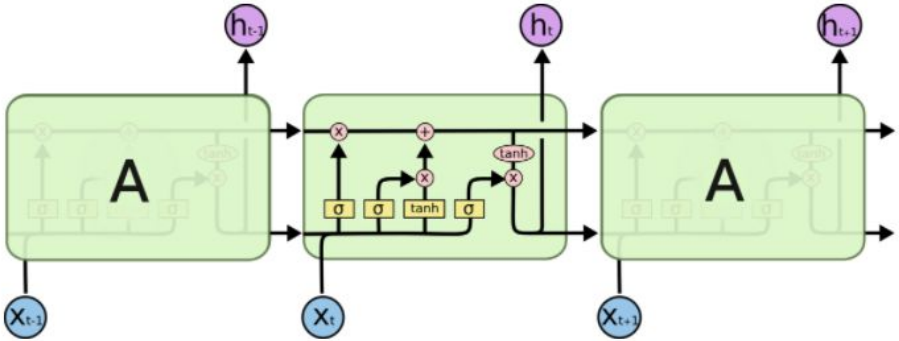
Result:

Investment period	Final money using algorithm 1	Final money using algorithm 2
2010 - 2012	18811	15668
2012 - 2014	10029	13275
2014 - 2016	13521	11724

Deep Learning

Utilize LSTM in Text Classification

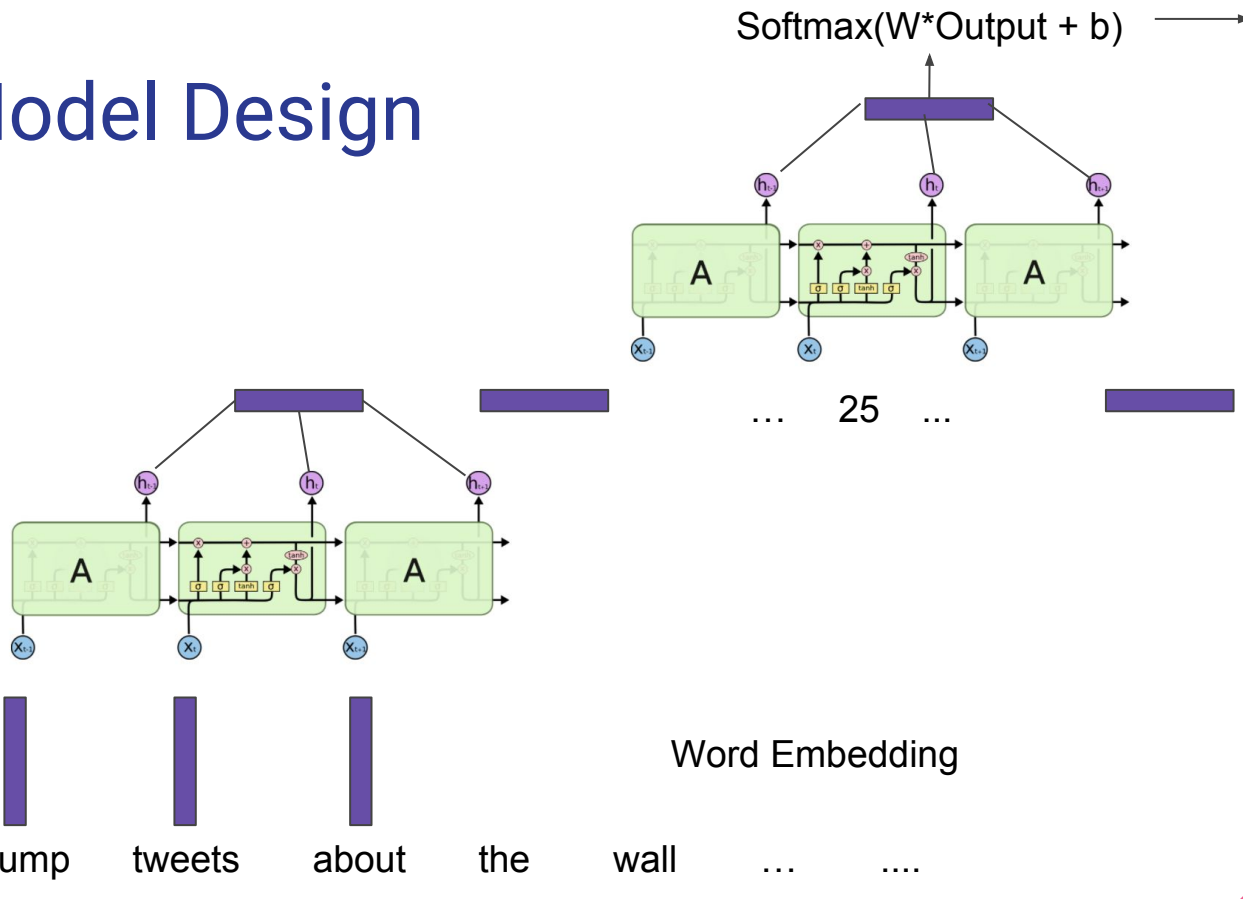
- Language features lost when using traditional approaches like n-grams
- LSTM
  - Sequential structure
  - Keep long memory
  - Begin-to-end hidden state
- LSTM is very suitable to process text!



Utilize Hierarchical LSTM for Long Text

- 25 headlines for each day
  - Sentences to long!
  - Single LSTM is not enough to process all the features in 25 headlines
- A solution to represent long document is needed for this task

Model Design



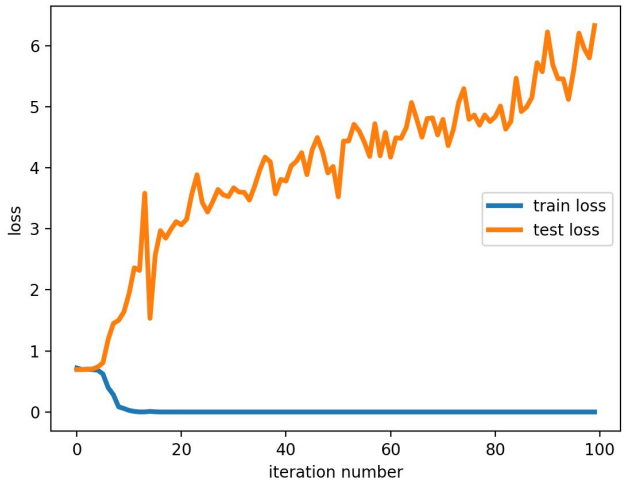
Code at: https://github.com/derekmma/stock-pred

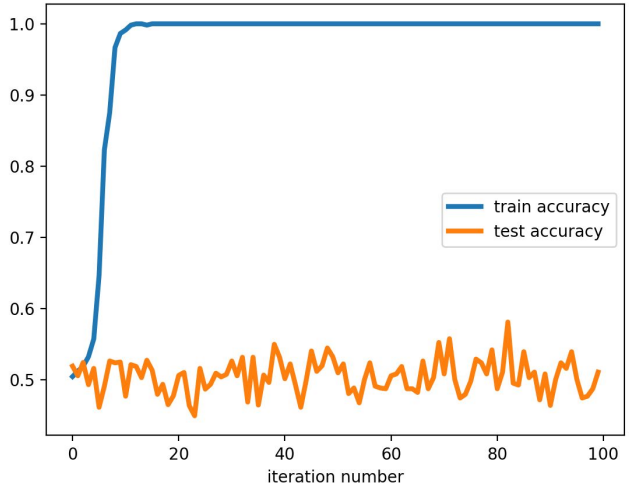
Word Embedding

- Bag of Words (BoW)
  - Lots of info will be lost
- Each word embed to pre-trained word vectors
  - GloVe (Pennington et al. 2014) from Stanford
  - Trained usign Twitter and Wikipedia
- Version
  - To speed up training process, use simpliest version
    - Wikipedia 2014 and Gigaword 5
    - 400K vocabulary
    - 50 dimension size

Evaluation and Analysis

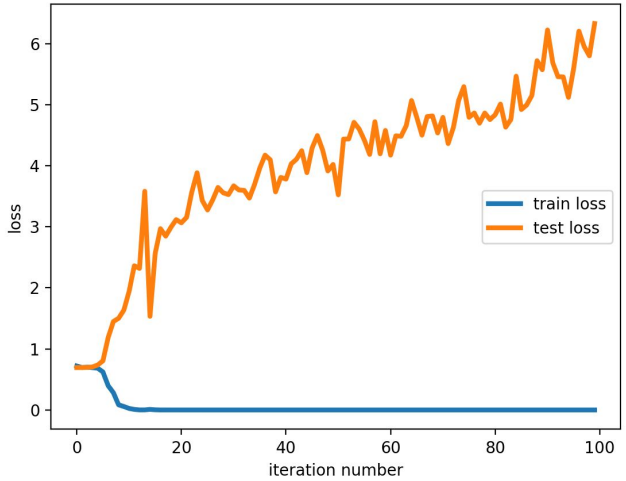
- Best accuracy is 0.581
- After around 10 iterations in total 100 iterations
  - Train loss is very small; train accuracy is very high
  - Test loss is increasing; test accuracy varies

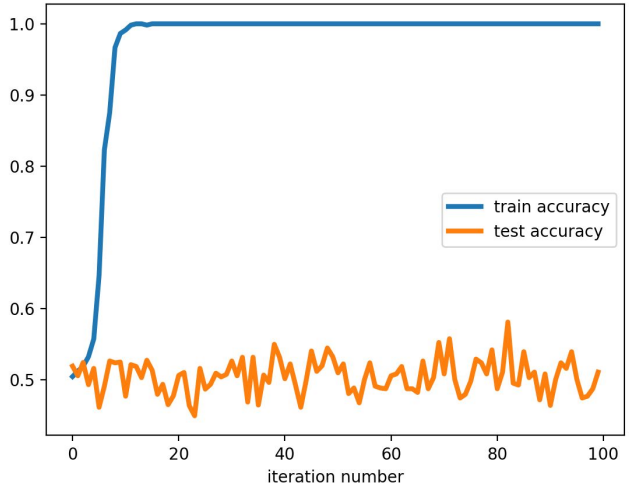




Evaluation and Analysis

- Over-fitting issue
- Possible solution
  - SImplify the model
  - Add more data sources





Conclusion

Conclusion

- Traditional Language Feature Methods
  - NN Multi-layer perceptron classifier SGD performs the best
  - 0.56 accuracy
- No clear clue to show that data cleaning by CoreNLP is benefitial
- N-day shift can improve the prediction accuracy
- Deep Learning Method
  - Hierarchical LSTM achieves 0.581 accuracy
  - Over-fitting problem is serious
- An automatic trading bot is developed
  - Buy-all-sell-all strategy
  - Probability cautious strategy