STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

Sentiment Analysis for Online Shopping

-Amazon Products

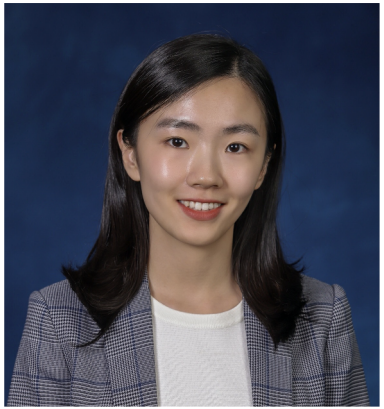
Course: BIA-678A
Team: Team-1

Professor: David Belanger

Data :12/14/2020

Members of Team 1

BIA-678-A

Zhoutao Cao	Ling Ao	Xuan Zhao	Yanran Min
			
Data Science	BIA Program	BIA Program	BIA Program
China	China	China	China

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

PRESENTATION

AGENDA

1. Introduction
2. Data Collection & Processing
3. Methodology
4. Model Comparison & Evaluation
5. Conclusion

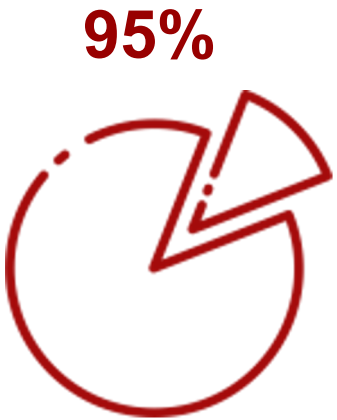
Introduction_Zhoutao

How Amazon Review Stars Work

Introduction

Objective

Which model is the best for sentiment analysis of Amazon product reviews?



Introduction

Framework

- Data Understanding
- Model Selection
- AWS Setup
- Model Evaluation
- Local & AWS Comparison

Introduction

Framework

- Data Understanding
- Model Selection
- Colab Setup
- Model Evaluation
- Local & Cloud Comparison

Data Collection & Processing_Ling Ao

Data Source: Amazon Review Data (2018)

- The dataset contains all reviews about 27 categories, total 233.1 million, in the range of May 1996 to Oct 2018.

- **Two Subsets:** CDs_and_Vintl dataset: 4,543,369 reviews
  Office Products dataset: 5,581,313 reviews

- **Columns:** reviewer ID, asin (ID of the product), reviewer Name, vote (helpful votes), style, reviewText, overall rating, summary, unixreviewTime, reviewTime, image

Data Collection & Processing

Label the Reviews

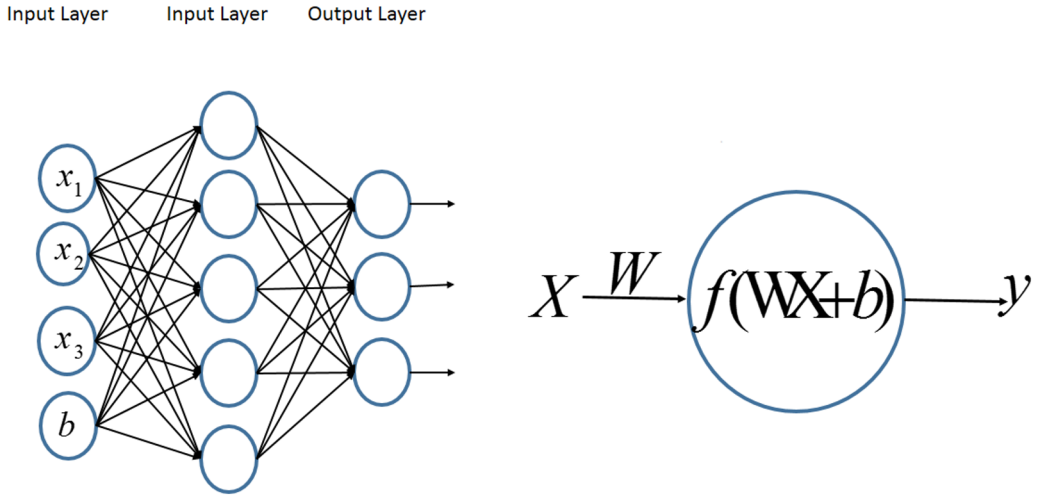
Negative Reviews
(Rating 1-3)

Positive Reviews
(Rating 4-5)

Methodology_Zhoutao

- Data size
  - LSTM, Multi-Layer Perceptron (MLP), Naive bayes and Logistic regression
- Model Chosen
  - 100,1000,5000,10000,100000,200000,400000,800000,1200000,1600000

Methodology - MLP Model_Zhoutao

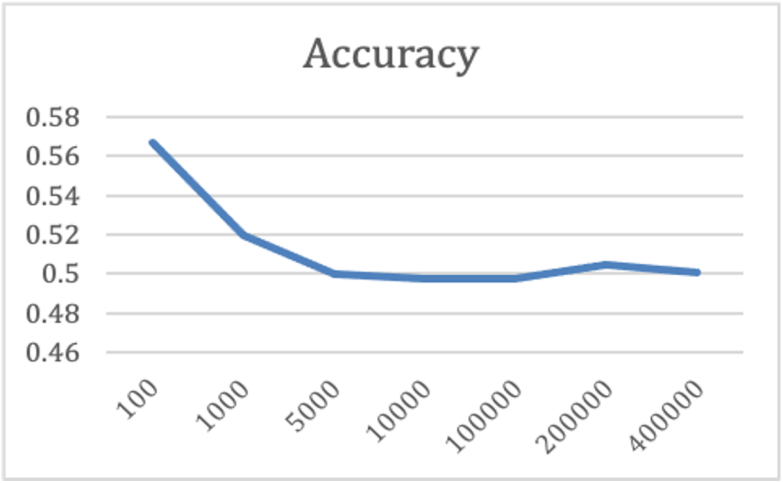


Methodology - MLP Model

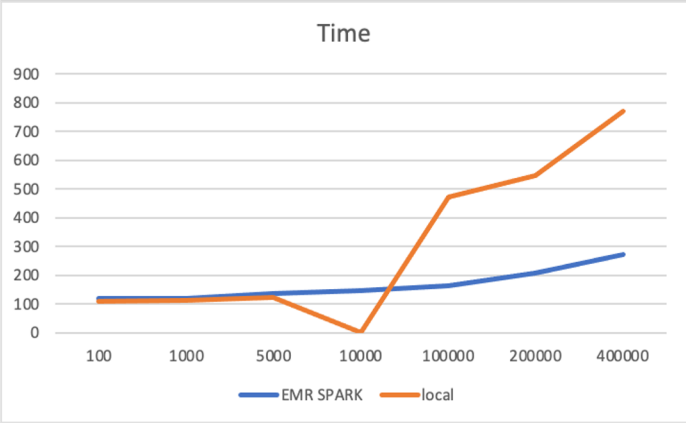
- Data cleaning
  - We randomly choose the data with 50 percent negative and 50 percent positive.
  - Our choices of data size are 100,1000,5000,10000,100000,200000,400000.

label	0	1	2	3	4	5	6	7	8	9	...	90	91	92	93	94	95	96	97	98	99
1.0	215	10	42	215	10	27	23	1	62	1015	...	0	0	0	0	0	0	0	0	0	0
1.0	1	285	8	22	3	1	165	244	11	1	...	13	2	14	44	7	72	85	79	47	3
1.0	292	1	187	11	1056	2	81	4	23	10	...	0	0	0	0	0	0	0	0	0	0
0.0	74	22	53	48	45	15	10	17	45	6	...	0	0	0	0	0	0	0	0	0	0
1.0	1059	1060	2	5	247	454	3	1061	20	80	...	218	29	5	1078	1079	1080	168	650	40	75
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1.0	2905	458	2906	17	1	62	390	924	2907	2908	...	0	0	0	0	0	0	0	0	0	0
0.0	6	118	24	368	12	7	145	29	2913	62	...	0	0	0	0	0	0	0	0	0	0
0.0	532	7	522	3	1	1002	2917	125	17	5	...	0	0	0	0	0	0	0	0	0	0
0.0	7	8	437	24	1	62	2920	3	335	2921	...	161	102	2933	51	439	412	34	2934	38	894
0.0	60	3	32	6	218	2937	12	6	39	1	...	9	1	257	324	31	2950	2	346	943	6

Methodology - MLP Model



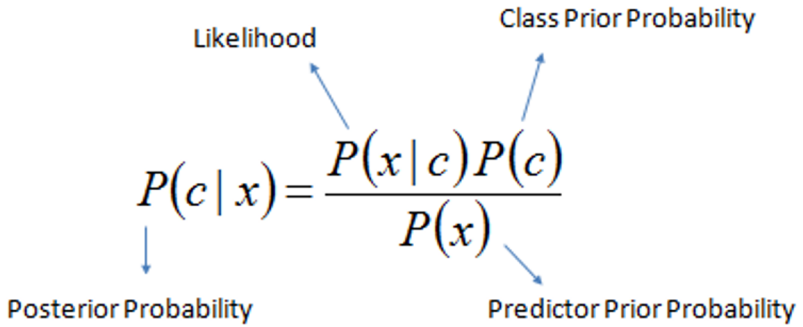
Methodology - MLP Model



By comparing, we found that if we use the data less than 10000, the personal computer is better than AWS. When the data size increases, the personal computer looks like it uses much more time than AWS.

Methodology - Naive Bayes Model_Xuan

- Why we used this model?
  - Simplicity: Less training time
  - Great for text classification: Treat each word as a unique feature



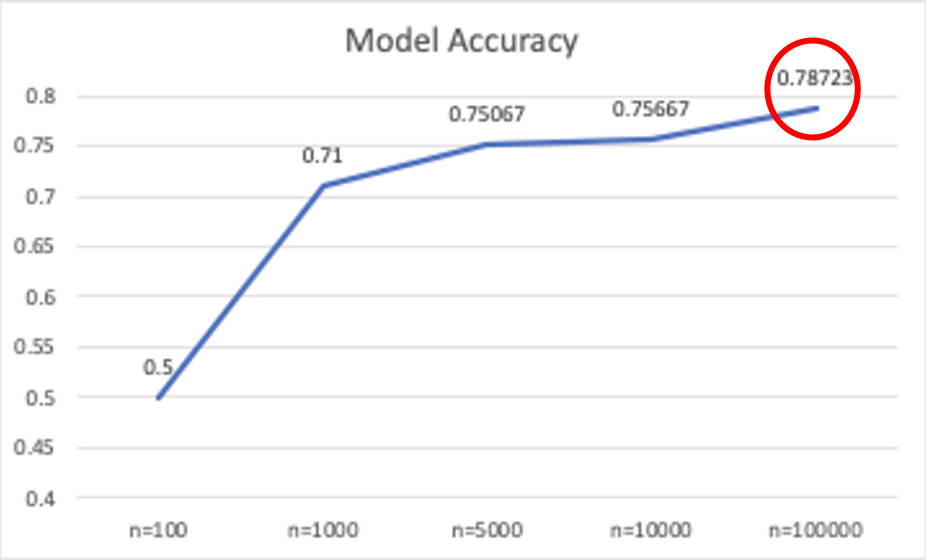
$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Methodology - Naive Bayes Model

- Model Preparation
  - Divide the ratings into positive (label = 1) and negative (label = 0)
  - Add a new column “cleaned_reviews”: remove stop words, punctuations, tokenize text in the “text” column
  - 70% of training data, 30% of testing data
  - Using TF-IDF to create two dataframes: count_df and tfidf_df

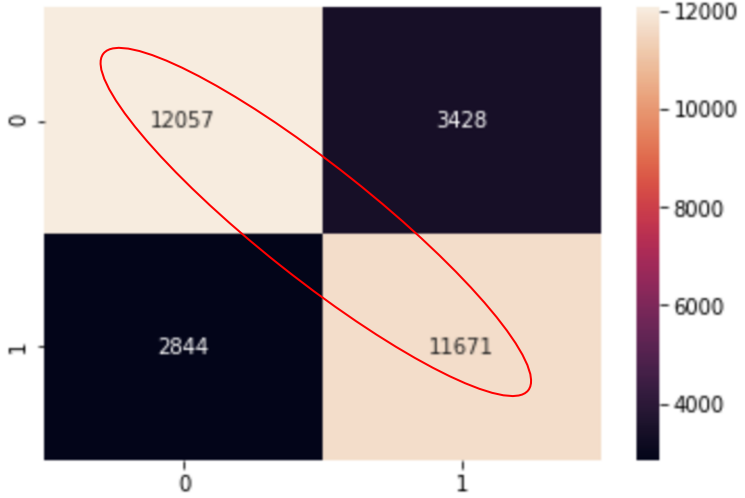
	text	label	cleaned_reviews
288852	let me mention the positive aspects first the ...	0.0	let mention positive aspects first cello prelu...
351392	i go look at the cd here on amazoncom and the ...	0.0	go look cd amazoncom reviews awsome get cd rea...
372751	i cant believe it says they wrote over 150 son...	0.0	cant believe says wrote 150 songs album wonder...
104173	2012 proved to be a horrible of year in the mu...	1	2012 proved horrible year music industry artis...
242385	this could be a good equation right dans spac...	0	could good equation right dans spacy aesthetic...

Methodology - Naive Bayes Model



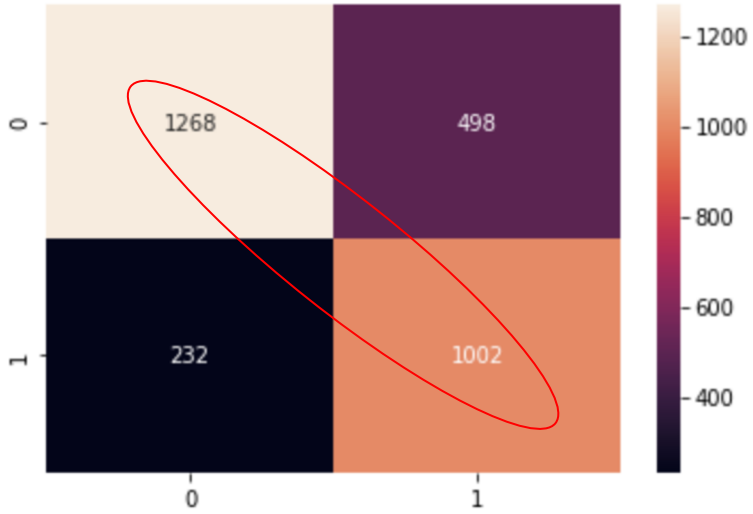
Accuracy goes up as sample size increases

Methodology - Naive Bayes Model



Sample n = 100,000

```
Accuracy: 0.7872333333333333
F score: 0.7872333333333333
```



Sample n = 10,000

```
Accuracy: 0.7566666666666667
F score: 0.7566666666666667
```

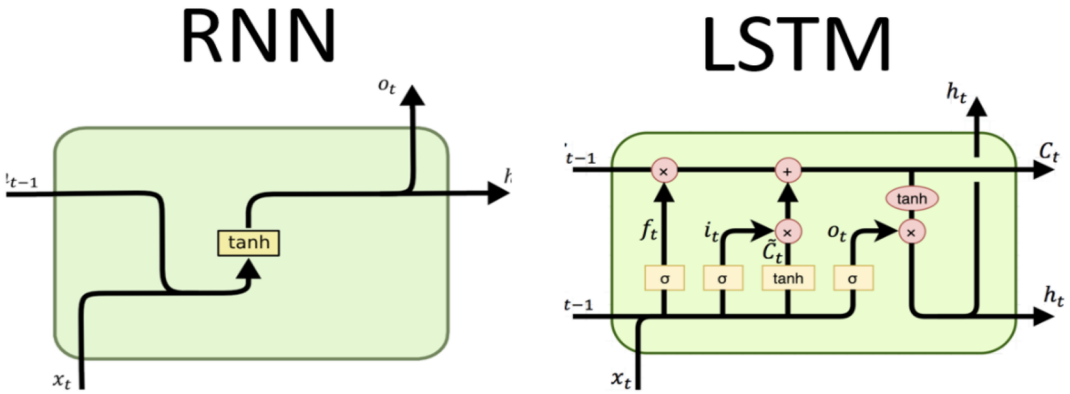
Methodology – LSTM_Ling Ao

- LSTM stands for long-short term memory
- Hochreiter Sepp and Schmidhuber Jurgen proposed LSTM in 1997
- A special recurrent neural network model used in deep learning

Methodology – LSTM

Why LSTM?

- ❑ Works better on long sequences due to the function of selecting memory.
- ❑ The average length of the reviews is 215



- More cells
- Tanh and sigmoid function

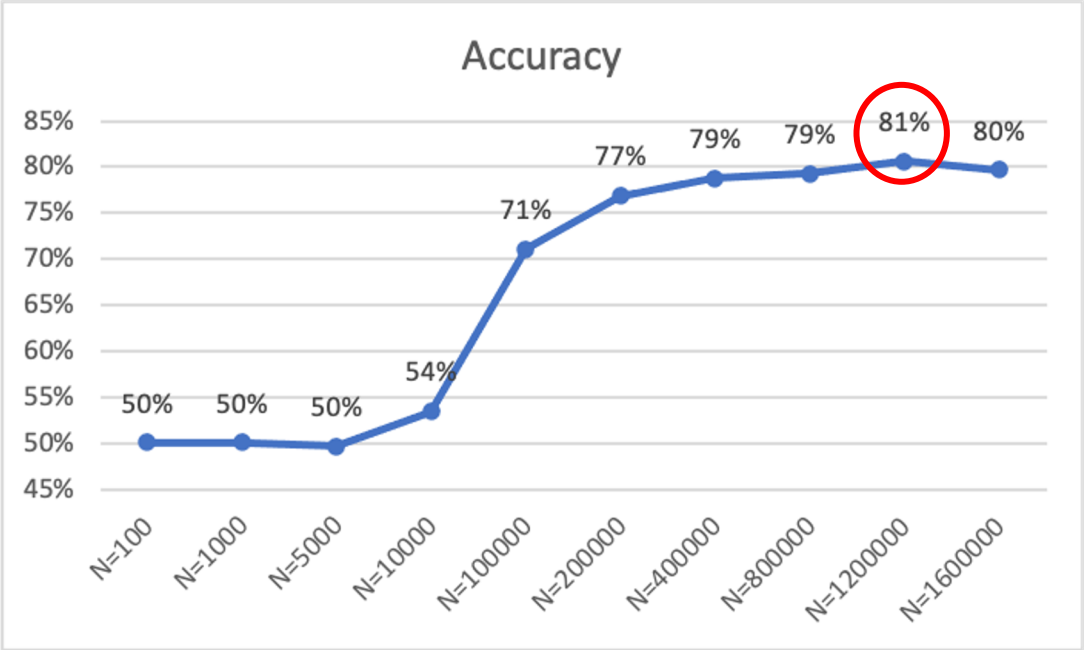
Methodology – LSTM

Model Parameters

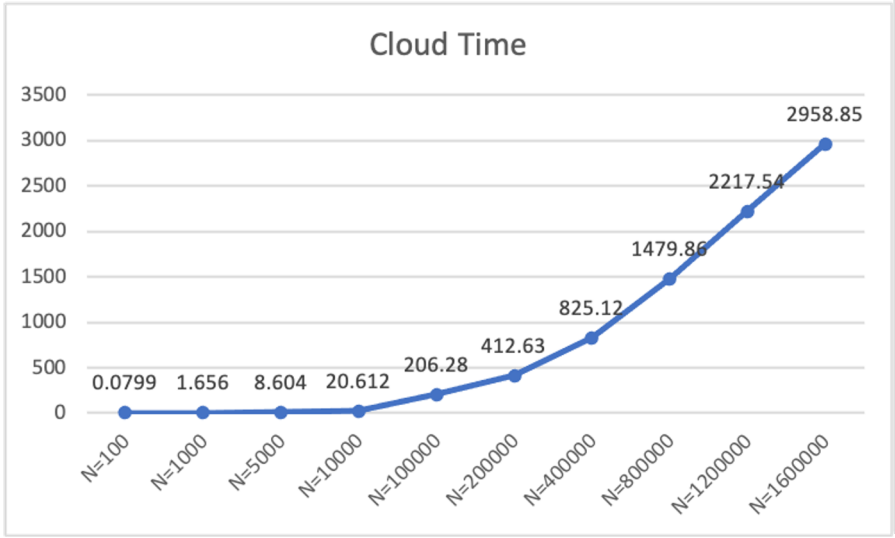
```
# Instantiate the model w/ hyperparams
vocab_size = len(vocab_to_int)+1 # +1 for the 0 padding + our word tokens
output_size = 1
embedding_dim = 200
hidden_dim = 128
n_layers = 3
```

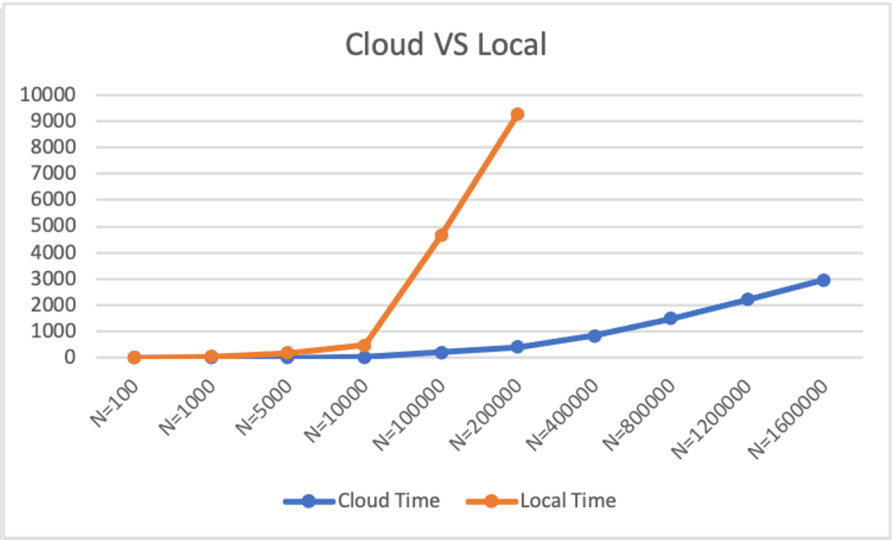
```
## Training
lr=0.0001
criterion = nn.BCELoss()
optimizer = torch.optim.Adam(net.parameters(), lr=lr)

epochs = 4
counter = 0
print_every = 100
clip= 500 # gradient clipping
max_step = 20000
```

Methodology – LSTM



Methodology – LSTM





Cloud time:Colab with GPU Local time: Colab with CPU

Methodology - Logistic Regression_Yanran

- Why Logistic Regression?

  - ❑ One of the simplest machine learning algorithms; easy to implement yet provides great training efficiency
  - ❑ Sometimes used as a benchmark model to measure performance
  - ❑ Very easy to scale
  - ❑ Requires less training time

Methodology - Logistic Regression

- Data Preprocessing

Similarly, We apply the same method as we did with Naive Bayes Model, where we adopt the **TF-IDF methodology**.

TD-IDF: a measure of originality of a word by comparing the # appearance of a word in a doc with the number of docs the word appear in.

```
tokenizer = Tokenizer(inputCol="review_text", outputCol="words")
hashtf = HashingTF(numFeatures=2**16, inputCol="words", outputCol='tf')
idf = IDF(inputCol='tf', outputCol="features", minDocFreq=5)
label_stringIdx = StringIndexer(inputCol="overall", outputCol="label")
pipeline = Pipeline(stages=[tokenizer, hashtf, idf, label_stringIdx])

pipelineFit = pipeline.fit(train_set)
train_df = pipelineFit.transform(train_set)
```

```
train_df.tail(1)
```

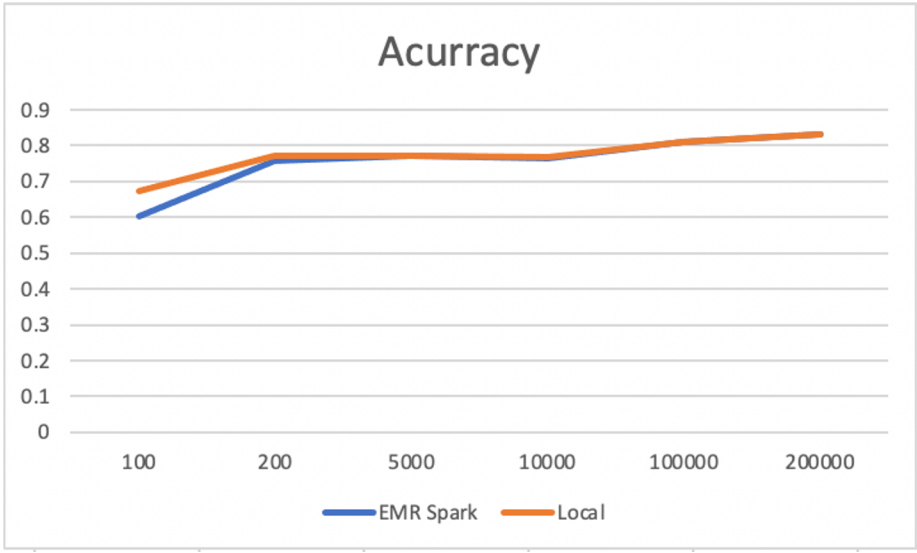
```
[Row(review_text='works well  Even ink flow  Good for leaving a bold impression  I tried this nib as a replacement for the Fine nib on my Expert 2 pen  It fits perfectly and works well  but I definitely prefer the Fine nib  This nib is not a good choice for individuals with a small  fine  or delicate writing style ', overall=1, words=['works', 'well', '', '', 'even', 'ink', 'flow', '', '', 'good', 'for', 'leaving', 'a', 'bold', 'impression', '', '', 'i', 'tried', 'this', 'nib', 'as', 'a', 'replacement', 'for', 'the', 'fine', 'nib', 'on', 'my', 'expert', '2', 'pen', '', '', 'it', 'fits', 'perfectly', 'and', 'works', 'well', '', 'but', 'i', 'definitely', 'prefer', 'the', 'fine', 'nib', '', '', 'this', 'nib', 'is', 'not', 'a', 'good', 'choice', 'for', 'individuals', 'with', 'a', 'small', '', 'fine', '', 'or', 'delicate', 'writing', 'style'], tf=SparseVector(65536, {1880: 1.0, 1981: 1.0, 3121: 1.0, 3232: 1.0, 4398: 1.0, 5809: 1.0, 6346: 1.0, 9781: 1.0, 12524: 1.0, 15945: 1.0, 19036: 2.0, 23307: 1.0, 25085: 1.0, 26143: 1.0, 26231: 1.0, 28302: 1.0, 28726: 1.0, 30138: 1.0, 30353: 2.0, 30950: 1.0, 33358: 1.0, 33917: 1.0, 37923: 1.0, 38269: 1.0, 39379: 1.0, 41240: 3.0, 41305: 1.0, 41571: 4.0, 43005: 2.0, 43894: 1.0, 44336: 1.0, 47896: 2.0, 48584: 4.0, 49343: 3.0, 50249: 2.0, 50418: 1.0, 52572: 13.0, 53917: 1.0, 55853: 2.0, 59317: 1.0, 60930: 1.0, 62234: 1.0}), features=SparseVector(65536, {1880: 1.0237, 1981: 1.359, 3121: 4.2164, 3232: 4.9719, 4398: 5.5068, 5809: 5.2314, 6346: 3.5785, 9781: 3.4335, 12524: 2.7236, 15945: 1.8835, 19036: 0.6094, 23307: 0.2955, 25085: 0.8862, 26143: 3.4838, 26231: 3.6331, 28302: 2.9159, 28726: 0.0, 30138: 3.1897, 30353: 0.3819, 30950: 0.4394, 33358: 2.8268, 33917: 0.9361, 37923: 6.0849, 38269: 4.8897, 39379: 5.9623, 41240: 1.8988, 41305: 0.5963, 41571: 1.5204, 43005: 1.1237, 43894: 2.2092, 44336: 0.9075, 47896: 3.3468, 48584: 22.6602, 49343: 7.8985, 50249: 4.3791, 50418: 6.5813, 52572: 0.6032, 53917: 1.7154, 55853: 3.8753, 59317: 4.9719, 60930: 0.9646, 62234: 7.2745}), label=1.0)]
```

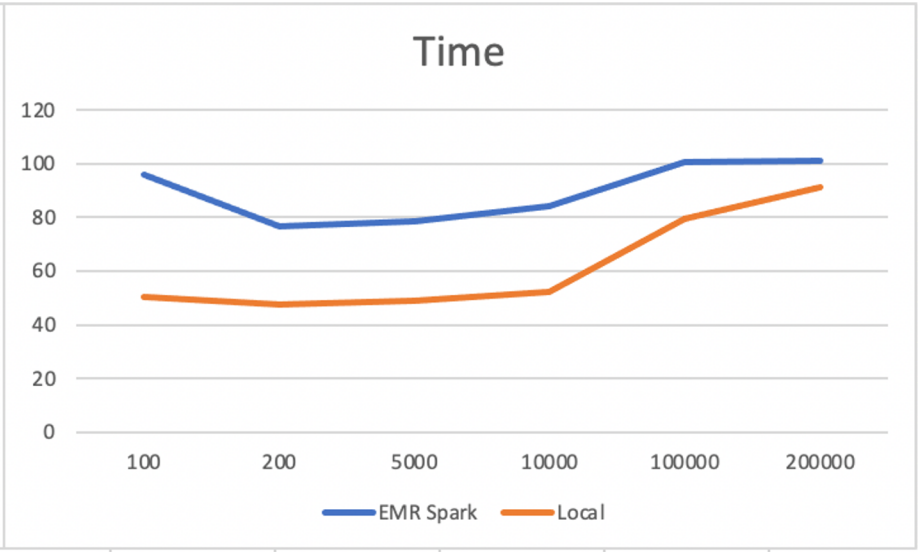
Methodology - Logistic Regression

- Split the data into training sets and testing sets
- Make the training data set of size 100, 1000, 5000, 10000, 100000, 200000, 400000 and the testing sets adjusted accordingly to make a 70/30 ratio
- Experiment on both local and AWS for the exact same script

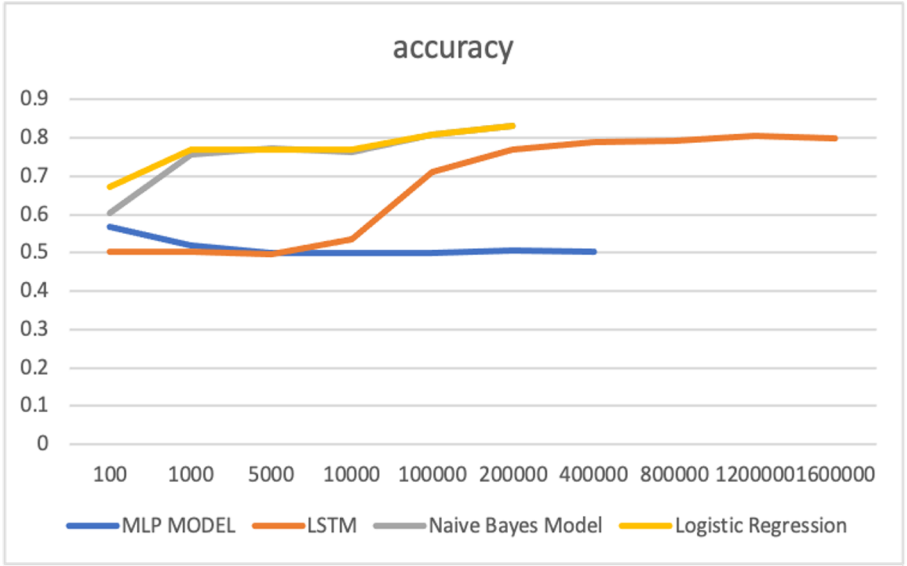
Methodology - Logistic Regression

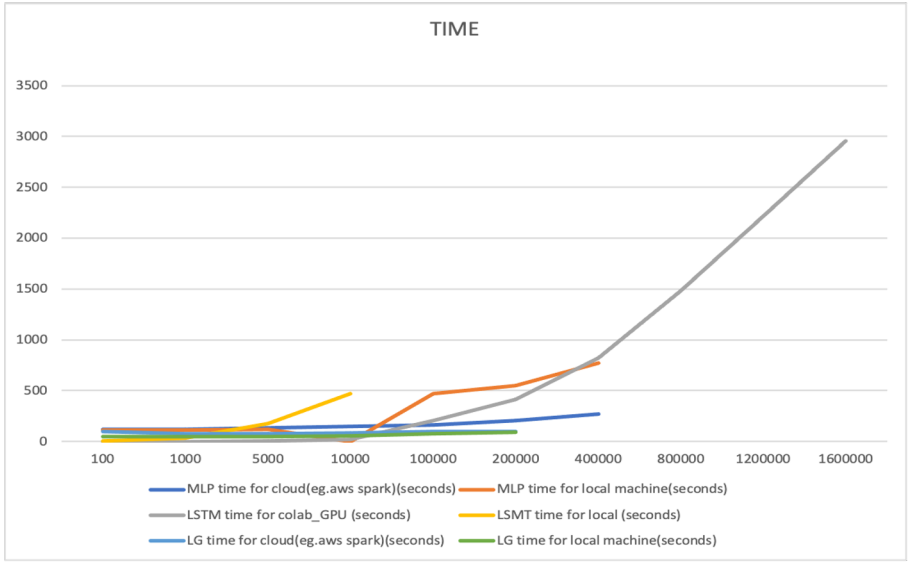
- Train the model locally and on AWS





Model Comparison & Evaluation_Yanran





Conclusion

- For the speed part, the speed of logistic regression is the fastest one.
- For the accuracy part, we found that the MLP has the lowest accuracy of 50%
- As a result, if using a small data size to train a model, Logistic regression and Naive Bayes Model are the first choices. They perform faster and generate higher accuracy. If the data size is big enough, we should choose LSTM for training.

Thank you!

Reference

Bhatt, A., Patel, A., Chheda, H., & Gawande, K. (2015). Amazon review classification and *International Journal of Computer Science and Information Technologies*, *6*(6), 5107-5110.

Brownlee, J. (2019, August 19). When to Use MLP, CNN, and RNN Neural Networks. Retrieved December 11, 2020, from https://machinelearningmastery.com/when-to use-mlp-cnn-and-rnn-neural-networks/

Data description: https://nijianmo.github.io/amazon/index.html#sample-review

Haque, T. U., Saber, N. N., & Shah, F. M. (2018, May). Sentiment analysis on large scale Amazon product reviews. In 2018 IEEE International Conference on Innovative Research and Development (ICIRD) (pp. 1-6). IEEE.

Hiroshi, K., Tetsuya, N., & Hideo, W. (2004, August). Deeper sentiment analysis using machine translation technology. In *Proceedings of the 20th international conference on Computational Linguistics* (p. 494). Association for Computational Linguistics.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8),1735-1780.

Reference

McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, No. 1, pp. 41-48).

Nasukawa, T., & Yi, J. (2003, October). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70-77).

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*.

Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, *3*(2), 143-157.

Wang, J., Yu, L. C., Lai, K. R., & Zhang, X. (2016, August). Dimensional sentiment analysis using a regional CNN-LSTM model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 225-230).

Reference

Wang, X., Liu, Y., Sun, C. J., Wang, B., & Wang, X. (2015, July). Predicting polarities of tweets by composing word embeddings with long short-term memory. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 1343-1353).

Yi, J., Nasukawa, T., Bunescu, R., & Niblack, W. (2003, November). Sentiment analyzer:Extracting sentiments about a given topic using natural language processing techniques. In *Third IEEE international conference on data mining* (pp. 427-434). IEEE.

STEVENS

INSTITUTE *of* TECHNOLOGY

THE INNOVATION UNIVERSITY®

stevens.edu

Zhuotao Cao, Ling Ao, Xuan Zhao, Yanran Min