W266 Final Project Report

Ehsan Yousefzadeh

Summer 2018

Abstract

To implement a neural network and naive bayes model to a data set of whiskey reviews in order to predict the price of a particular bottle of whiskey using a binary classifier (0 for cheap and 1 for expensive)

Introduction

The project will focus on the whiskey reviews data set released by the Whiskey Advocate that is available on Kaggle. The data set includes 22,000 entries with the following features:

- name: Name of whisky bottle
- category: Whisky category
- review.point: Point marked by each reviewers
- price: Price of each bottle
- currency: Unit of price
- description: Descriptions of reviews

The purpose of the project is to predict the price of a bottle (either expensive or cheap) based on any user review description. The model will need to address to main areas: sentiment analysis of the review and key words in the review. For instance, if someone writes a positive review and mentions certain flavor notes, such as caramel and peat, these words could be associated with bottles that sell at a higher price. This model will help whiskey makers better understand what flavor profiles are important for determining the price of a bottle. The models to be used will be a Naive Bayes model or a Neural Net LSTM model that uses binary classification to determine the final price (expensive or cheap) of a bottle which is a binary output. The model's goal would be to minimize the softmax cross entropy loss of the logits that are being outputted.

Methods

The methods I will be using will be to prepare my data set for the models then implement the two models I have chosen. The baseline model is a Naive Bayes model and the main model will be a neural network using tensorflow and deep learning. The baseline model will be a simple Naive Bayes implementation using the scikit learn standard library. Our neural network model will be a tensorflow implementation with the following components:

- **Embedding layer:** $x^{(i)} = W_{embed}[w^{(i)}]$
- **Summing vectors:** $x = \sum_{i=1}^{n} x^{(i)}$
- **Hidden layer(s):** $h^{(j)} = f(h^{(j-1)}W^{(j)} + b^{(j)})$ where $h^{(-1)} = x$ and $j = 0, 1, \ldots, J-1$
- **Output layer:** $\hat{y} = \hat{P}(y) = \text{softmax}(h^{(final)}W_{out} + b_{out})$ where $h^{(final)} = h^{(J-1)}$ is the output of the last hidden layer.

Our data set will first need to be loaded and prepared in order to fit the model's parameters. To do this, the raw text data will have to be tokenized, canonicalized, converted to IDs corresponding to the full vocabulary dictionary, then padded with a "0" constant values so that each of the data examples have a uniform vector shape. Once this is complete, the data can then be properly loaded into our embedding layer and run through the neural network to then output the binary classification price of the bottle.

Results and Discussion

After running the dataset in our baseline Naive Bayes model, we were able to get an accuracy of 59.22% on our test dataset. This is relatively low given that our dataset is binarized. After running the same test dataset in our neural network model, we are able to achieve a 68.61% accuracy -- an improvement of 15.86%. We can see from below on Figure 1 that the model was able to successfully minimize the cross-entropy loss. The accuracy of the model began to increase over training time however we see that the accuracy began to slightly dip -- indicating that the model was overfitting. After adjusting some parameters, I was running into the same

problem with overfitting. I came to the conclusion that this had to do with the fact that my dataset was too small and the model was beginning to memorize it too easily.

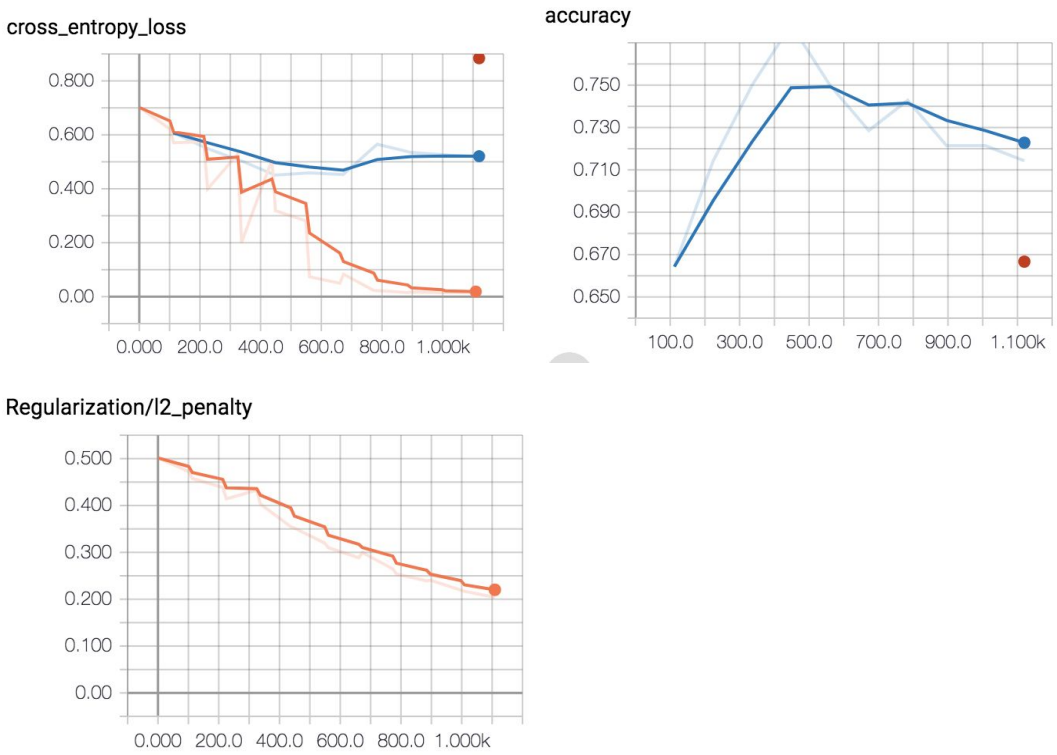

Figure 1

After performing an error analysis on the data we received the following (Figure 2):

```
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.68      0.73       204
           1       0.51      0.67      0.58       105

avg / total       0.70      0.67      0.68       309

Confusion Matrix:
 [[138  66]
 [ 35  70]]
```

Figure 2

We can see that the precision for the 0s (cheap bottles) was much higher than the precision of the 1s (expensive bottles), 0.80 vs 0.51 respectively. This means that we were able to successfully predict 80% of the cheap bottles and 51% of the expensive bottles. Meanwhile, the

recall of each classification is roughly the same at 0.68 and 0.67. The model had a much easier time classifying the 0s (cheap bottles) than the 1s (expensive bottles).

After visually examining examples of 0s that were classified as 1s and 20 examples of 1s that were classified as 0s, I noticed a couple of observations. First, a lot of the 1s that were classified as 0s contained a reference to other more expensive bottles and were comparing them in its own class. Here is an example:

```
Misclassified as 0 that should be a 1:
 The only distillery-approved bottling of Bladnoch in the United States. When compared to the Gordon & MacPhail bottl
ing above, this one isn't as sweet or creamy in texture. There's more citrus fruit and a drier, spicier finish. This
one is also bolder and a bit more aggressive.
```

As you can see from this example, you can see that a key focus of this review is its reference to another expensive brand called Bladnoch. If you were to take out the reference to the other label, you would interpret this review as being one of a cheaper priced bottle. It seems as though when a review references other high-end labels, then the sentiment of the review is adjusted downward since they already alluded to it having a high standard in reference to another expensive label. Given the limited usage of words referencing other brands and expensive labels, our model failed to give proper weighting to those words. This could be significantly improved with a larger data set, thus when other labels are mentioned it will give a much higher weighting to the example. The second observation I have is that it was increasingly difficult to tell why the model tended to misclassify. Upon reading many examples, I myself could not ascertain whether the bottle was cheap or expensive. I think the world of fine scotch is incredibly complex where the price of a bottle is not just dictated by the review of the scotch, but rather it is also influenced by factors outside of text analysis such as the bottle's rarity and availability. Thus, a model that incorporates both text analysis and other variables would be much stronger than analyzing language alone.

Conclusion

The goal of this project was to analyze scotch review data and determine if a bottle was cheap (0 classification) or expensive (1 classification). Upon preparing the data, we were able to successfully run a baseline model (Naive Bayes using Sci-kit learn) and the final model (Neural Network using TensorFlow). The accuracy for our baseline model was 59% whereas the

accuracy for our final model was 69%. Upon analyzing the results, we saw that classifying the 1s was a more challenging task than predicting the 0s. Further analysis into the actual text revealed that certain references to other expensive bottles was overlooked in our model, which could be solvable given a larger dataset. Lastly, the world of fine scotch is just as complex as the scotch itself. Deducing prices based solely on text will yield limited results as factors such as rarity and availability are overlooked. A model that incorporates both text and other factors will be far more superior.

Additional Notes

I had originally intended to create a neural network with an output layer that is a linear regression per the below:

- **Embedding layer:** $x^{(i)} = W_{embed}[w^{(i)}]$
- **Summing vectors:** $x = \sum_{i=1}^{n} x^{(i)}$
- **Hidden layer(s):** $h^{(j)} = f(h^{(j-1)}W^{(j)} + b^{(j)})$ where $h^{(-1)} = x$ and $j = 0, 1, \ldots, J-1$
- **Output layer:** $\hat{y} = \hat{P}(y) = (h^{(final)}W_{out} + b_{out})$ where $h^{(final)} = h^{(J-1)}$ is the output of the last hidden layer.

This model uses a cost function that minimizes the root mean squared error (RMSE) of the labels and the predicted outputs. After setting up the model, the results were inconclusive as the output was yielding a constant value for all of our predictions. If I had additional time I would be able to troubleshoot the model to see why it was generating a constant value. The code for this model is provided as a supplement.

References

1. Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C. (2018). *A Neural Probabilistic Language Model*. [online] Jmlr.org. Available at: http://www.jmlr.org/papers/v3/bengio03a.html [Accessed 17 Jun. 2018].
2. 2. https://dl.acm.org/citation.cfm?id=1935932
3. https://www.sciencedirect.com/science/article/pii/S0957417413008518
4. https://dl.acm.org/citation.cfm?id=1014073
5. https://www.kaggle.com/koki25ando/22000-scotch-whisky-reviews
6. W266 Course Material (UC Berkeley School of Information)