MAIS 202 - PROJECT DELIVERABLE 3

1. Final Training Results

Recall that the objective of this project is to predict a price category for a wine, given it's description in the form of a review.

Unfortunately, I did not have the chance to implement a better model. I tried for a long period of time to implement a neural network, but in the end it does not perform better than other models like the SVM. I would need to tweak more parameters and/or implement a recurrent neural network, more suitable for NLP. I am still trying to implement the RNN, but the release of the web app will take precedence over venturing into new models.

I have refined the data processing to remove any number from the data fed into the model(s). I felt that it could improve accuracy, but it did not have any effect on the accuracy. I have chosen to present the results of the Support Vector Machine model and the Random Forest classifier as to keep the discussion concise. Out of all models, the RFC offers the best results by a margin of a few percent points. Below are the results of accuracy, precision, recall and confusion matrices for the two models:

```
The test accuracy of the Linear SVC model is 55.65382707885601%
The train set accuracy of the Linear SVC model is 61.35088218815727%
The precision of the SVM model is [0.63779164 0.52217742 0.50036792 0.2375     0.14285714 0.5       ]
The recall of the SVM model is [0.6986336  0.55543641 0.49026676 0.03362832 0.00877193 0.06666667]

-----------------------------------------

The test accuracy of the RFC model is 59.91072904612332%
The train set accuracy of the RFC model is 99.99081532371392%
The precision of the RFC model is [0.6810664  0.5383835  0.59970958 0.98387097 1.         1.        ]
The recall of the RFC model is [0.69812753 0.68282222 0.44664744 0.1079646  0.03508772 0.03333333]
SVM confusion matrix:
[[2761 1068  122    1    0    0]
 [1269 2590  789   14    0    1]
 [ 269 1105 1360   36    4    0]
 [  26  166  352   19    1    1]
 [   3   25   78    7    1    0]
 [   1    6   17    3    1    2]]
RFC confusion matrix:
[[2759 1127   66    0    0    0]
 [1044 3184  435    0    0    0]
 [ 214 1320 1239    1    0    0]
 [  28  228  248   61    0    0]
 [   6   38   66    0    4    0]
 [   0   17   12    0    0    1]]
```

I note that the confusion matrices show that the models have a lot of trouble differentiating the $0-20 and $20-40 category. Further tests merging the two categories have shown that the accuracy is increased to over 85% if the two categories are merged. This opens for a number of explanations, one of which being that the model(s) cannot differentiate between the two categories because the differences are too subtle, or that perhaps the wines in the two categories are of equivalent "taste value." In other words, it would mean that the price is not correlated to the taste for these categories.

As for the precision of the models, it seems like they are more precise in the lower classes (but this is to be expected since 70% of the data points fall in classes 1 and 2). The precision and recall of the RFC are overall higher than the SVM.

The SVM beats the RFC only for classes 3 and 7, but not overall because these classes represent a small percentage of the total data. I find surprising that even if the RFC seems to be overfitting the data (seen in the train set accuracy), it ends up generalizing slightly more than 3% better on the test set than the SVM. The RFC also tended to underestimate the prices of the wines more than the SVM. We can see this in the confusion matrix, because the entries are more stacked to the left in the SVM confusion matrix.

2. Final demonstration proposal

The final product will be very simple. The user will input a description of a wine, taken from an announce or a website. Then, the webapp will output the predicted price category of the wine, from $0-20, $20-40, $40-80, $80-150, $150-300, $300-1000 or more than $1000. The interface will consist of a field where the user can input the description and a button “predict price.”

I have no experience whatsoever with web development. I will start with a simple Flask application and stylesheet, but I might try to create a React app from it if time permits. To develop my Flask app, I will rewatch the workshop and implement the architecture presented. If need be, I will watch youtube tutorials and read blog posts. I will also find a website presenting a quick introduction to css to improve the visual aspect of my website.

Most importantly, I believe that this project can be helpful for people who eat out and want to figure out the true cost of the wine they are drinking. Based on the description from the menu, the application can predict the price category and inform the user if the restaurant price is too expensive or within an acceptable range from what the wine is truly worth.

To finish, I see two potential improvements to the model. First, a recurrent neural network could be implemented to try to increase the accuracy of the model. Second, the input could be changed from a text to the picture of a text, which would greatly improve the user experience. To do so, I would implement a convolutional neural network to convert the image to text, and then feed this into the recurrent neural network.