MBAN 6500 A Assignment 3 -- Sentiment Classification

For this assignment you will build a sentiment classifier to classify restaurant reviews.

Submission

This assignment should be submitted as Python 3 code and uploaded to Canvas. The submission should be a single `PY` file, and **not** a Jupyter Notebook. The due date for the first part is on April 5 at 11:59 pm.

The code will be tested and should produce the output specified below.

You will be using a the file `reviews.csv` that contains close to 2000 reviews. Values in `reviews.csv` are tab separated instead of comma separated as if often the case in `CSV` files. To read `reviews.csv` with `pandas` it might be good to specify tab as the delimiter, as follows:

```
data = pd.read_csv('reviews.csv', delimiter='\t')
```

or

```
data = pd.read_csv('reviews.csv', sep='\t')
```

The two are equivalent.

Task

Your task is to do sentiment analysis of the reviews. You will do this by training and testing a BoW text classifier on the review texts using `RatingValue` as labels. The ratings should be binned into negative (ratings 1 & 2), neutral (rating 3) and positive (ratings 4 & 5) sentiment.

However, the ratings are very unbalanced, there are many more positive (4 & 5) ratings than negative (1 & 2) ratings. You will need to drop positive ratings in order to balance the data so that you have approximately equal numbers of negative, neutral and positive ratings.

Split the data into training and validation sets and store them as `training.csv` and `valid.csv`. The validation data should be used for model selection and evaluation.

For this task it is important that you report performance carefully using both accuracy, F1-score and a confusion matrix.

Deliverable

You need to submit a single Python (`PY` **NOT** `IPYNB`), that does the following:

- Loads the `reviews.csv`, preprocesses the data, splits it and saves the files as `training.csv` and `valid.csv`.
- Loads `training.csv` and trains the model.
- Loads the validation data (`valid.csv`) and prints the performance metrics on the validation set.
    - I.e. it should print something like the following:

```
accuracy: "accuracy on the test set"

F1_score: "f1-score on the test set"

Confusion_matrix:
        negative neutral positive
negative   a       b        c
neutral    d       e        f
positive   g       h        i
```

Grading

For grading the validation data (`valid.csv`) will be replaced with unseen test data (`test.csv`) and for full marks your model needs to perform above chance.

Good luck!