MMAI 5400 Assignment 2 -- 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 is on February 22 at 8:30 am.

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

Data

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. The binned ratings should be coded with negative as `0`, neutral as `1` and positive as `2`. This new column should be called `Sentiment`. See **Table 1** below for a description of the format.

However, the ratings are very unbalanced, there are many more positive (`2`) ratings than negative (`0`) 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.

The resulting table should look as follows:

Table 1

	Sentiment	Review
	`1`	`"It shows ..."`
	`0`	`"Disgusting..."`
	`2`	`"Yummy..."`
…	…	…

Once this is done, you should split the data into training and validation sets and save them as `train.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:

1. Loads the `reviews.csv`, preprocesses the data, splits it and saves the files as `train.csv` and `valid.csv`.
2. Loads `train.csv` and trains the model.
3. 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`). `test.csv` has the same format as described by **Table 1**, and as `train.csv` and `valid.csv` should have.

For full marks the following conditions have to be met.

- The file can only contain Python code (i.e. no `pip install` satements).
- File paths should be specificed at the top of the file, right below the import statements.
- F1-score should use `macro`.
- Plot the confusion matrix with pairwise F1-scores.
- All F1-scores have to be greater than 0.
- No google colab specific code (the code with be tested on a local machine, not on google colab).
- Your model needs to perform above chance on an unseen test set (`test.csv`).

Good luck!