p17jayavignesh@iimnagpur.ac.in

[Email address]

Jayavignesh

P17027

p17jayavignesh@iimnagpur.ac.in

Text mining and Social media analytics

Option 1 assignment(coding)

Code Results and explanation:

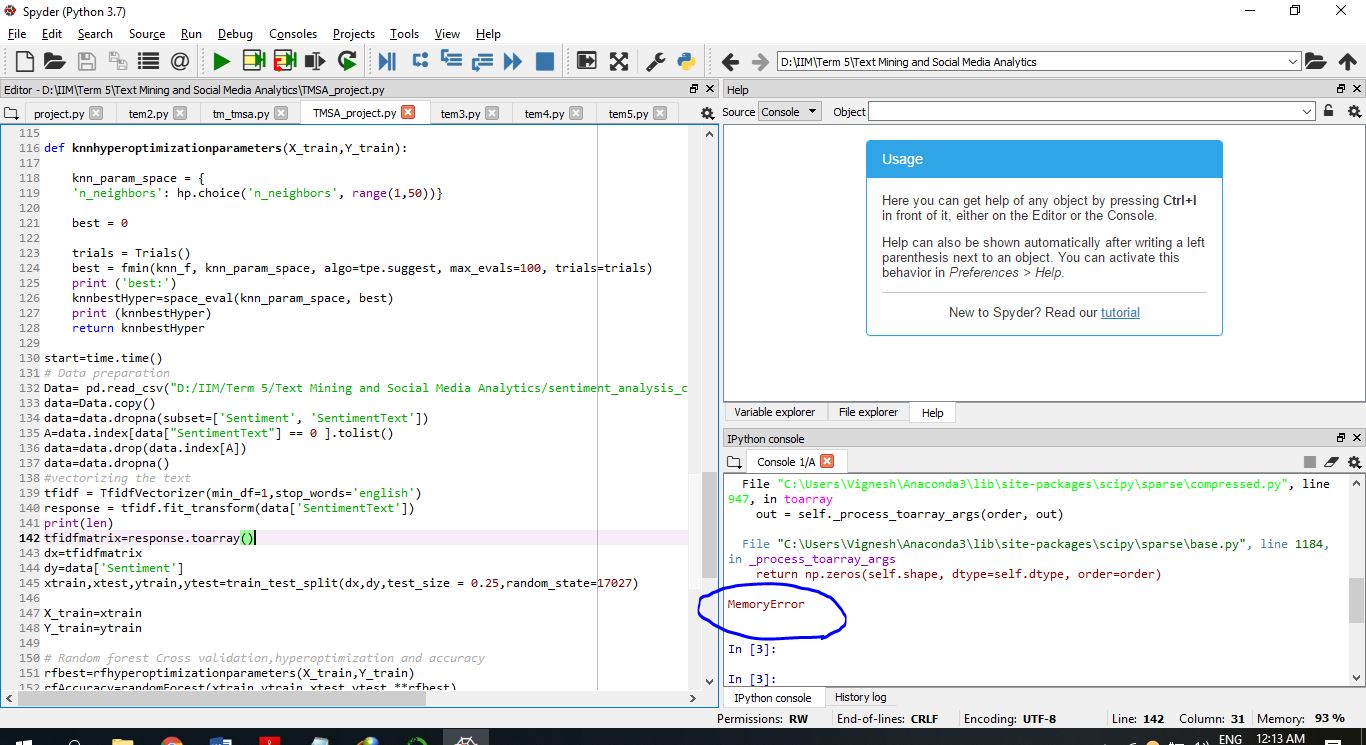
# Code walkthrough:

The data is stored in the data frame using pandas. The data is then cleaned from unwanted characters like null. The documents in the data are parsed and then vectorized into a sparse matrix tfidf algorithm. The vectorized matrix is the independent variable data and the corresponding sentiment is the dependent variable data. Now the test and train data is split using a 25% test train split method. The training data is fed to three models (SVM, KNN, Random forest). All the three models do hyperparameter tuning by Bayesian optimization methods and the fine-tuned parameters are used for training and testing the data set. This process is repeated for all the three models and the hold out models scores are printed.

# Data length:

As my computer is able to predict only handle 10000 rows of data due to memory error Iam using only the 10000 rows.

# Proof of memory error:



Code working requirement: All the libraries and modules should be installed in python and the file path should be changed to the file location in the system which is going to run the code. If memory error pops up then the no. of documents (i.e., no. of rows should be reduced). The datafile with .csv extension has a encoding problem so I copied the first 10000 rows of text data into the new .xlsx file version. Due to my laptop lack of processing power I kept the Bayesian hyper parameter tuning fmin function max evaluation to 1.

# Code results:

Total time taken for execution in seconds : 1.3681480884552002

new best: 0.6810975014163199 {'criterion': 'entropy', 'max\_depth': 1, 'max\_features': 50, 'n\_estimators': 318}

best:

{'criterion': 'entropy', 'max\_depth': 1, 'max\_features': 50, 'n\_estimators': 318}

Random forest Accuracy 0.6834437086092715

new best: 0.7643663677743265 {'C': 1.339727041600196, 'gamma': 0.2976267816190692, 'kernel': 'linear'}

best:

{'C': 1.339727041600196, 'gamma': 0.2976267816190692, 'kernel': 'linear'}

SVM Accuracy 0.7675496688741722

new best: 0.680656000312567 {'n\_neighbors': 22}

best:

{'n\_neighbors': 22}

KNN Accuracy 0.6834437086092715

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Random forest Best hyper parameters

{'criterion': 'entropy', 'max\_depth': 1, 'max\_features': 50, 'n\_estimators': 318}

Support vector Best hyper parameters

{'C': 1.339727041600196, 'gamma': 0.2976267816190692, 'kernel': 'linear'}

KNN Best hyper parameters

{'n\_neighbors': 22}

Random forest Accuracy= 0.6834437086092715

Support Vector Machine Accuracy= 0.7675496688741722

K-nearest neighbors Accuracy= 0.6834437086092715

Total time taken for execution in seconds : 2436.994956970215

Total time taken for execution in hr : 0.6769433212942547

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# References:

https://nlpforhackers.io/tf-idf/ tokenizer

https://towardsdatascience.com/automated-machine-learning-hyperparameter-tuning-in-python-dfda59b72f8a

https://github.com/hyperopt/hyperopt/wiki/FMin

https://github.com/WillKoehrsen/hyperparameter-optimization/blob/master/Bayesian%20Hyperparameter%20Optimization%20of%20Gradient%20Boosting%20Machine.ipynb

https://github.com/WillKoehrsen/hyperparameter-optimization/blob/master/Introduction%20to%20Bayesian%20Optimization%20with%20Hyperopt.ipynb

https://towardsdatascience.com/an-introductory-example-of-bayesian-optimization-in-python-with-hyperopt-aae40fff4ff0

https://github.com/ChenglongChen/Bayesian-Optimization-with-Gaussian-Processes/blob/master/examples/sklearn\_example.py

https://github.com/roamanalytics/roamresearch/blob/master/BlogPosts/Hyperparameter\_tuning\_comparison/hyperparameter\_tuning\_comparison\_code.py

http://steventhornton.ca/hyperparameter-tuning-with-hyperopt-in-python/

https://medium.com/district-data-labs/parameter-tuning-with-hyperopt-faa86acdfdce

https://www.kaggle.com/rpygamer/random-forest-from-grid-search-to-hyperopt

http://steventhornton.ca/hyperparameter-tuning-with-hyperopt-in-python/

https://medium.com/@ramrajchandradevan/comparison-among-hyper-parameter-optimizers-cd37483cd47

https://roamanalytics.com/2016/09/15/optimizing-the-hyperparameter-of-which-hyperparameter-optimizer-to-use/#Use-with-scikit-learn-models

https://github.com/roamanalytics/roamresearch/blob/master/BlogPosts/Hyperparameter\_tuning\_comparison/hyperparameter\_tuning\_comparison\_code.py

https://stackoverflow.com/questions/25006369/what-is-sklearn-cross-validation-cross-val-score

https://stackoverflow.com/questions/3282823/get-the-key-corresponding-to-the-minimum-value-within-a-dictionary

https://github.com/hyperopt/hyperopt/issues/284

https://github.com/hyperopt/hyperopt/wiki/FMin

https://medium.com/district-data-labs/parameter-tuning-with-hyperopt-faa86acdfdce

https://medium.com/@mikkokotila/a-comprehensive-list-of-hyperparameter-optimization-tuning-solutions-88e067f19d9

https://www.youtube.com/watch?v=bPYJi1E9xeM

https://www.youtube.com/watch?v=dA\_x2xHTYQE

https://www.youtube.com/watch?v=GofidHdilas

https://www.youtube.com/watch?v=m-S9Hojj1as

https://www.youtube.com/watch?v=m-S9Hojj1as

https://www.youtube.com/watch?v=GM3CDQfQ4sw

https://www.youtube.com/watch?v=ErDgauqnTHk

https://www.youtube.com/watch?v=nhqo0u1a6fw