End of year project data science

Yehudit Brickner

Explanation about the project

Part 1 - trying to do better than the last semesters classification
In this part of the project I started by taking the features with numerical values and dividing by 100, so that my data will be numbers between 0-1.
Next I split the data into train and test data sets and then normalized the features that were non-numeric by taking the average score per subgroup of each feature.
Then I changed the value to a float.
After that I split the data sets into x, y and started running the training models.
I ran XGBoost, AdaBoost, GradientBoost, and VotingClassifier.
I ran those models on 3 different datasets with different amounts of features.
My best result is 87.375% using XGBoost with the data set using all the features.

Part 2 - Fashion MNIST
I started by importing the data and splitting into train and test data sets.
Then I started running models with 3 different PCA values 0.9 = 84 features, 0.8 = 24 features, 0.7 = 9 features.
The models I used are KNN, XGBoost, Random Forest and VotingClassifier.
I did not normalize the data because it is already normalized. It's the value of the pixels in grey scale.
The best result I got, in comparison to the least amount of features, is 82.26% with PCA 0.7 = 9 features.

Part 3 - dogs Vs cats
The first thing I did was create a function to make the data into a csv file, and downloaded it to my computer. Once I downloaded the csv file, I commented out the function, so that it wouldn't run each time I reloaded the notebook.
Then I imported the data from the csv and divided all the data by 255, so that I would hopefully get better results. (That didn't work as planned, the results were the same.)
Next, I split the data into 3 parts train, test, and final test set. I started running models with 3 different PCA values, 0.9 = 331, 0.8 = 72, 0.7 = 25.
The modes I used are KNN, XGBoost, Random Forest, Logistic Regression, VotingClassifier, and for PCA 0.7 I ran a Pipeline with standard scaler using all the classifiers above except for VotingClassifier.
The best result I got, in comparison to the least amount of features, is 64.26% with PCA 0.7 = 9 features.

Part 4 - hands classification
The first thing I did was import all the data, then I started to slowly organize it.
I added the right hand so that it is next to the left hand. I removed rows that contain null values, and I took off the first 7 seconds from all the data. Lastly, I combined it all into the train data frame and shuffled it. I decided to use person 4 as my test data for the training.
Next I started training the models KNN, XGBoost, Random Forest, Logistic

Regression, AdaBoost, VotingClassifier and a Pipeline with standard scaler using KNN, XGBoost, GradientBoost, AdaBoost. And taking those results and creating my own classifier.
After that I imported the test data and repeated the steps to get it organized into a data frame and ran the tests.
The best result that I got is 87.9% using the pipeline with KNN.