Data Science Project

Improving the model from last semester:

In last semester's project I created a model to predict if someone will have a heart attack in the future or not. The data I used was imbalanced, almost all the cases were: " will not have a stroke" cases, my model predicted almost all the cases ass "will not have a stroke" and managed to get a high score. This time I used an ensemble XGBOOST model and added weight to the "will have a stroke" cases and managed to get the same score with detecting more of the "will have a stroke" cases correctly with also using PCA and using 9 features instead of 22.

Fashion Mnist:

Applied PCA to reduce dimensions and used 82 features instead of 787.Created some simple knn, randomforestclassifier, and xgboost models who got some good scores, I tried to tune the models with randomizedsearchcv (gridsearch took too long)but got worse results.Then I created a voting classifier with the previous models as the estimators and gave each one different weight based on their performance and ended up getting a higher score of 91%.

Dogs VS Cats:

Applied PCA to reduce dimensions and used 286 features instead of 4800.Created some simple knn, randomforestclassifier, and xgboost models who got some good scores, I tried to tune the models with randomizedsearchcv (gridsearch took too long)but got worse results. Then I created a voting classifier with the previous models as the estimators and gave each one different weight based on their performance and ended up getting a higher score of 66%. Then I created a new voting classifier without knn and put it into a bagging classifier using soft voting and got 67.4%

Hand Sync:

I loaded all the data and merged the training data with the right hand file. Scaled the data with standard scaler. Visualized the data in 3d scatter graphs. Then I used PCA to reduce the dimensions and went from 44 features to 10. Created some simple knn, randomforestclassifier, and xgboost models who got some good scores, I tried to tune the models

with randomizedsearchcv (gridsearch took too long)but got worse results. Then I created a voting classifier with the previous models as the estimators and gave each one different weight based on their performance and ended up getting a higher score of 87%.