Data Visualization Summary

By: Daniel Appel
ID: 207386699

The different methods of classifiers and of preprocessing used during the Data Visualization Final Course Project, in order of notebooks:

- Previous semester notebook: **Airline Satisfaction:**
    - **Preprocessing:** finished last semester, unchanged.
        - Tried to use PCA for performance improvement, unsuccessfully, reverted to the classifiers with no PCA.
        - Using K-Means for preprocessing of data
    - **Classification models:**
        - Dummy
        - Logistic Regression
        - KNN
        - Gaussian Naïve Bayes
        - Random Forests
        - AdaBoost
        - XGBoost
        - Voting (on all previous classifiers with different weights)
        - Bagging (using XGBoost)
        - Pasting (using XGBoost)
        - Stacking (using classifiers up to and including XGBoost) with XGBoost as final estimator
        - For comparison: XGBoost after PCA
        - For comparison: AdaBoost after PCA
        - XGBoost after using K-Means for preprocessing
    - **Classifier evaluation methods:**
        - Confusion matrices
        - Classification reports
        - ROC curve + AUC
        - Bar graphs
        - Dataframe for saving and presenting of results

- **Fashion MNIST:**
    - **Preprocessing:**
        - Importing of data from Keras
        - Flattening 3D array to 2D array
        - Removal of unneeded features (unneeded in my opinion)
        - Checking for NaN values
        - Data splitting into Train, Test and Validation sets
        - Removing duplicate values
        - Scaling
        - PCA
        - Using K-Means for preprocessing of data
    - **Classification models:**
        - Dummy
        - Logistic Regression
        - KNN
        - Gaussian Naïve Bayes
        - Random Forests
        - AdaBoost
        - XGBoost
        - Voting (on all previous classifiers with different weights)
        - Bagging (using XGBoost with less estimators to improve runtime)
        - Pasting (using XGBoost with less estimators to improve runtime)
        - Stacking (using classifiers up to and including XGBoost) with XGBoost as final estimator
        - Voting after using K-Means for preprocessing
    - **Classifier evaluation methods:**
        - Confusion matrices
        - Classification reports
        - Bar graphs
        - Dataframe for saving and presenting of results

- **Cats vs Dogs**
  - **Preprocessing**
    - Using OpenCV to view and experiment on images
    - Resizing images to uniform shape
    - Flattening 3D array to 2D array
    - Export/Import to .csv
    - Checking for NaN values
    - Removing duplicates
    - Adding labels to data
    - Data splitting into Train, Test and Validation sets
    - Scaling
    - PCA
    - Using K-Means for preprocessing of data
    - Doing the preprocessing twice: once for grayscale images, once for colour images
  - **Classification models:**
    - Dummy
    - Logistic Regression
    - KNN
    - Gaussian Naïve Bayes
    - Random Forests
    - AdaBoost
    - XGBoost
    - Voting (on all previous classifiers with different weights)
    - Bagging (using XGBoost with less estimators to improve runtime)
    - Pasting (using XGBoost with less estimators to improve runtime)
    - Stacking (using classifiers up to and including XGBoost) with XGBoost as final estimator
    - XGBoost after using K-Means for preprocessing
  - **Classifier evaluation methods:**
    - Confusion matrices
    - Classification reports
    - ROC curve + AUC
    - Bar graphs
    - Dataframe for saving and presenting of results

- <u>**Hand Positioning**</u>
    - **Preprocessing**
        - Combining data into lists of dataframes
        - Checking for NaN – in case of handRight, the NaN are removed
        - Removing spaces from column names for ease of use
        - Removing incorrect data
        - Resetting indexes of dataframes when needed
        - Removing first 7 seconds of each dataframe (for cleaner data)
        - Removing duplicates
        - Combining handRight with all dataframes labeled "alone"
        - Removing records where the same FrameID appears more than twice (for spont and sync)
        - Adding labels to data
        - Combining each two consecutive rows of data (in multiple ways)
        - Merging all dataframes into one
        - Export/Import dataframe to .csv for ease of use and runtime efficiency
        - Removing unneeded features
        - Siphoning data (taking every 5th row, as every single row would cause overfitting (more overfitting))
        - Using correlation graph to determine dependent features to be further reduced
        - Viewing data histograms for deeper understanding of data
        - Splitting into train, test and validation sets
        - PCA
        - Using K-Means for preprocessing of data
    - **Classification models:**
        - Dummy
        - Logistic Regression
        - KNN
        - Gaussian Naïve Bayes
        - Random Forests
        - AdaBoost
        - XGBoost
        - Voting (on all previous classifiers with different weights)
        - Bagging (using XGBoost with less estimators to improve runtime)
        - Pasting (using XGBoost with less estimators to improve runtime)
        - Stacking (using classifiers up to and including XGBoost) with XGBoost as final estimator
        - XGBoost after using K-Means for preprocessing
        - XGBoost after multiplication of data
    - **Classifier evaluation methods:**
        - Confusion matrices
        - Classification reports
        - ROC curve + AUC
        - Bar graphs

- Dataframe for saving and presenting of results