UbiComp Assignment-2

Name: Luka Bekavac
Date: 23.10.2023

1)

Different features, parameters and models were tested on the dataset. The best result of each batch trained with all Models () is summarized in following table.

Accuracy	Model	Feature Set	Normalizer
0.8961038961038961	Random Forest	#Features 1	maxAbs
0.8831168831168831	Random Forest	#Features 2	maxAbs
0.8831168831168831	GB-Classifier	#Features 3	maxAbs
0.8961038961038961	Random Forest	#Features 4	maxAbs
0.9090909090909091	Random Forest	#Features 1	standard
0.8831168831168831	Random Forest	#Features 2	standard
0.8831168831168831	Random Forest	#Features 3	standard
0.8961038961038961	Random Forest	#Features 4	standard

I could reach an accuracy of **0.909**
I decided to tweak following parameters:

a) Feature-engineering

During the feature engineering phase, I experimented with the number of features used in the model. I iterated through four distinct sets of features to determine which combination would yield the best results. The features used in each round of testing are as follows:

- New Features Round 4:
  In this iteration, I used the following features:

  df[['meanFix', 'maxFix', 'varFix', 'stdFix', 'meanDis', 'varDis', 'stdDisp', 'freqDisPerSec', 'xDir', 'yDir', 'fixDensPerBB', 'number_of_blinks', 'blinkRate', 'blinkMean']]

- New Features Round 3:
  For the third round, I used these features:

  df[['meanFix', 'maxFix', 'varFix', 'stdFix', 'meanDis', 'varDis', 'stdDisp', 'freqDisPerSec', 'xDir', 'yDir', 'fixDensPerBB']]

- New Features Round 2:
  In this set, I just used all of the features available:

  df[['meanFix', 'minFix', 'maxFix', 'varFix', 'stdFix', 'meanDis', 'minDis', 'maxDis', 'varDis', 'stdDisp', 'freqDisPerSec', 'number_of_blinks', 'blinkMean', 'blinkMin', 'blinkMax', 'blinkRate', 'xDir', 'yDir', 'fixDensPerBB']]

- New Feature Set 1:
  This was the initial set of features which was already provided which I tested:

  df[["meanFix", "maxFix", "varFix", "xDir", "yDir", "fixDensPerBB"]]

I chose the features based on following thoughts.

1. **Fixation Metrics (e.g., meanFix, maxFix, varFix, stdFix):**
   Longer fixations are common in reading, varied durations in searching, and a mix in inspection.
2. **Displacement Metrics (e.g., meanDis, varDis, stdDisp, freqDisPerSec):**
   Reading involves systematic movements, searching has frequent shifts, and inspection focuses on specific areas.
3. **Directional Metrics (xDir, yDir):**
   Reading shows horizontal eye movements, searching involves multi-directional scans, and inspection might have patterned movements.
4. **Fixation Density (fixDensPerBB):**
   High density in reading indicates interest, in searching it points to potential information locations, and in inspection it highlights areas of focus.
5. **Blink Metrics (e.g., number_of_blinks, blinkRate, blinkMean):**
   Frequent blinking suggests cognitive load in searching, while reduced blinking in reading indicates concentration. Inspection blink patterns vary based on the inspected item's complexity.
6. **Minimum and Maximum Values (e.g., minFix, maxFix, minDis, maxDis):**
   The maximum and minimum values of fixations and displacements can help differentiate activities based on duration and focus intensity.

b) Normalizing

To preprocess and standardize the feature set for better model performance, I used two different normalization techniques:

Max Absolute Scaler (given):
This method scales each feature by its maximum absolute value. It's particularly useful when the data contains both positive and negative values and ensures that the scaled data falls within the range of -1 to 1. After applying the MaxAbsScaler, the features were normalized, as can be observed in the displayed subset of the "Normalized Features."

Standard Scaler:
The StandardScaler was employed to standardize the feature set. This scaler removes the mean and scales the features to unit variance. Essentially, it transforms the data such that the distribution has a mean value of 0 and a standard deviation of 1. The results of this scaling can be viewed in the "Standardized Features" subset.

Both normalization techniques aim to scale the feature values to a similar range, ensuring that no particular feature dominates the model due to its scale or variance, thus aiding in more accurate and efficient training.

Depending on the model, both normalizing methods proved to be improving the accuracy of the model, compared to no normalizing.

c) Model selection

I added 2 more Models which I used in testing:

Random Forest
This proved to be the best performing model in almost all cases.

GradientBoostingClassifier
Another model which I thought could lead to good results.

Both models were chosen based on their ability to handle complex datasets and deliver robust predictions, making them ideal candidates for testing against the provided dataset.