Machine Learning
Exercise 2 - More Comparative Evaluation

Student: se21m024, Thomas Stummer

The source code can be found in the document ***se21m024_Stummer_ml_ex2_comp_eval.ipynb***.

Small data set: Heart Failure Prediction
The data set was provided by Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020) (https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-020-1023-5) and downloaded from https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data.

Big data set: Covertype
The data set was provided by Jock A. Blackard and Colorado State University and downloaded from https://archive.ics.uci.edu/ml/datasets/Covertype.

Music data set
This dataset contains 1000 songs (100 songs for 10 genres) and was downloaded from Moodle.

Small data set: Heart Failure Prediction

The data was split into input features a target feature. The target feature is 'DEATH_EVENT' that indicates either the person has died. The column 'time' was not used as input feature due to the direct connection to the target feature 'death_event' according to https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data/discussion/178372. The train/test split was chosen to be 2/3 to 1/3 as required. For the k-NN apporach, k-d tree was chosen as algorithm to gain results within a reasonable amount of time.

Results table

Algorithm with parameters	Accuracy	F1	Training time	Testing time
Perceptron (alpha: 0.0001)	0.394	0.223	0.002998 sec	0.001003 sec
Perceptron (alpha: 0.001)	0.394	0.223	0.002004 sec	0.001999 sec
Perceptron (alpha: 0.01)	0.394	0.223	0.00196 sec	0.001 sec
k-NN (1-NN)	0.566	0.532	0.003003 sec	0.004989 sec
k-NN (2-NN)	0.566	0.467	0.002 sec	0.004999 sec
k-NN (3-NN)	0.525	0.481	0.002003 sec	0.004998 sec
Decision Tree (max features: None)	0.616	0.594	0.002999 sec	0.000999 sec
Decision Tree (max features: sqrt)	0.697	0.689	0.002999 sec	0.001003 sec
Decision Tree (max features: log2)	0.697	0.689	0.002 sec	0.002003 sec
SVM	0.636	0.578	0.006001 sec	0.002001 sec
Random Forests (num trees: 10, max features: sqrt)	0.636	0.593	0.016998 sec	0.002001 sec
Random Forests (num trees: 10, max features: log2)	0.636	0.593	0.015 sec	0.002999 sec
Random Forests (num trees: 100, max features: sqrt)	0.687	0.653	0.128 sec	0.010965 sec
Random Forests (num trees: 100, max features: log2)	0.687	0.653	0.12086 sec	0.009999 sec

Interpretation

Accuracy and F1 measure

The highest (= best) accuracy and F1 measures were accomplished by the decision tree with the sqrt or the log2 function to limit the number of features for consideration for each split. An F1 measure of 0.689 was achieved. The worst accuracy and F1 measures were reached by the perceptron (the chosen alpha value had no impact on the outcome). An F1 measure of only 0.223 was achieved.

Training and testing time

The training time was very similar for the perceptron, the k-NN and the decision tree, ranging from about 0.002 seconds to about 0.003 seconds. With 0.006 seconds, the training time for the support vector machine (SVM) took about two to three times as long as the algorithms mentioned above. The training time for the random forests ranged from 0.015 seconds (with 10 trees) to 0.128 seconds (with 100 trees). It can be seen that the overall training took about 5.8 times as long when training 10 times more forests.

The testing time for the percepton was the least with 0.001 seconds an the longest for the random forests with about 0.011 seconds.

Big data set: Covertype

The data was split into input features and a target feature. The target feature is 'Forest cover type class' than can be any value between 1 and 7 and indicates which type of vegetation is growing there mainly. The train/test split was chosen to be 2/3 to 1/3 as required. For the k-NN approach, k-d tree was chosen as algorithm to gain results within a reasonable amount of time.

Results table

Algorithm with parameters	Accuracy	F1	Training time	Testing time
Perceptron (alpha: 0.0001)	0.527	0.439	0.082004 sec	0.001982 sec
Perceptron (alpha: 0.001)	0.527	0.439	0.115 sec	0.003002 sec
Perceptron (alpha: 0.01)	0.527	0.439	0.106003 sec	0.002 sec
k-NN (1-NN)	0.755	0.755	0.134 sec	0.180002 sec
k-NN (2-NN)	0.729	0.723	0.089997 sec	0.172999 sec
k-NN (3-NN)	0.735	0.731	0.09252 sec	0.167 sec
Decision Tree (max features: None)	0.708	0.71	0.073037 sec	0.001999 sec
Decision Tree (max features: sqrt)	0.688	0.69	0.016003 sec	0.001998 sec
Decision Tree (max features: log2)	0.673	0.677	0.014034 sec	0.002001 sec
SVM	0.746	0.736	2.287948 sec	0.912415 sec
Random Forests (num trees: 10, max features: sqrt)	0.782	0.777	0.086034 sec	0.008001 sec
Random Forests (num trees: 10, max features: log2)	0.772	0.768	0.079967 sec	0.009035 sec
Random Forests (num trees: 100, max features: sqrt)	0.802	0.798	0.925223 sec	0.063999 sec
Random Forests (num trees: 100, max features: log2)	0.801	0.796	0.790013 sec	0.067001 sec

Interpretation

Accuracy and F1 measure

The highest (= best) accuracy and F1 measures were accomplished by the random forests with 100 trees and the sqrt function to limit the number of features for consideration for each split. An F1 measure of 0.798 was achieved. Even when using only 10 trees the random forests outperformed all the other algorithms. The worst accuracy and F1 measures were reached by the perceptron (the chosen alpha value had no impact on the outcome). An F1 measure of only 0.439 was achieved.

Training and testing time

The training time was quite similar for the perceptron, the k-NN, the unlimited decision tree and the random forests with 10 trees, ranging from about 0.073 seconds to about 0.134 seconds. The shortest training time was achived by the descition tree with either the sqrt or the log2 limit function with 0.016 seconds resp. 0.014 seconds. With 2.288 seconds, the training time for the SVM was by far the most time consuming one. The training time for the random forests with 100 trees ranged from 0.79 seconds to 0.93 seconds and is located between the timespans mentioned above. Like in the small data set, it can be seen that the overall training took for random forests took about 10 times longer when training 10 times more forests.

With about 0.002 seconds, the testing time for the percepton and the decision tree was the least. The highest training time was required by the SVM with about 0.912 seconds.

Music data set

Results tables

Beats per minutes

Algorithm with parameters	Accuracy	F1	Training time	Testing time
Perceptron (alpha: 0.0001)	0.082	0.012	0.006035 sec	0.001002 sec
Perceptron (alpha: 0.001)	0.082	0.012	0.004999 sec	0.0 sec
Perceptron (alpha: 0.01)	0.082	0.012	0.005002 sec	0.000999 sec
k-NN (1-NN)	0.148	0.119	0.002 sec	0.007002 sec
k-NN (2-NN)	0.127	0.083	0.001 sec	0.006999 sec
k-NN (3-NN)	0.103	0.078	0.000999 sec	0.008002 sec
Decision Tree (max features: None)	0.173	0.134	0.001005 sec	0.000997 sec
Decision Tree (max features: sqrt)	0.173	0.134	0.001 sec	0.001001 sec
Decision Tree (max features: log2)	0.173	0.134	0.000999 sec	0.001 sec
SVM	0.179	0.098	0.017004 sec	0.004996 sec
Random Forests (num trees: 10, max features: sqrt)	0.167	0.129	0.013001 sec	0.001003 sec
Random Forests (num trees: 10, max features: log2)	0.167	0.129	0.012001 sec	0.001999 sec
Random Forests (num trees: 100, max features: sqrt)	0.164	0.124	0.110754 sec	0.009999 sec
Random Forests (num trees: 100, max features: log2)	0.164	0.124	0.111004 sec	0.01 sec

Beats per minutes statistics

Algorithm with parameters	Accuracy	F1	Training time	Testing time
Perceptron (alpha: 0.0001)	0.088	0.014	0.006968 sec	0.0 sec
Perceptron (alpha: 0.001)	0.088	0.014	0.006 sec	0.001 sec
Perceptron (alpha: 0.01)	0.088	0.014	0.007 sec	0.0 sec
k-NN (1-NN)	0.176	0.176	0.002035 sec	0.007997 sec
k-NN (2-NN)	0.13	0.123	0.001997 sec	0.008962 sec
k-NN (3-NN)	0.155	0.147	0.002034 sec	0.009 sec

Algorithm with parameters	Accuracy	F1	Training time	Testing time
Decision Tree (max features: None)	0.17	0.165	0.005042 sec	0.0 sec
Decision Tree (max features: sqrt)	0.2	0.198	0.002999 sec	0.0 sec
Decision Tree (max features: log2)	0.203	0.201	0.002999 sec	0.001001 sec
SVM	0.239	0.214	0.022999 sec	0.006 sec
Random Forests (num trees: 10, max features: sqrt)	0.212	0.203	0.019 sec	0.001999 sec
Random Forests (num trees: 10, max features: log2)	0.227	0.216	0.019999 sec	0.002001 sec
Random Forests (num trees: 100, max features: sqrt)	0.215	0.209	0.169999 sec	0.013 sec
Random Forests (num trees: 100, max features: log2)	0.212	0.207	0.191 sec	0.013001 sec

Chroma (similar to actual notes)

Algorithm with parameters	Accuracy	F1	Training time	Testing time
Perceptron (alpha: 0.0001)	0.303	0.275	0.015 sec	0.001001 sec
Perceptron (alpha: 0.001)	0.303	0.275	0.015 sec	0.0 sec
Perceptron (alpha: 0.01)	0.303	0.275	0.015 sec	0.0 sec
k-NN (1-NN)	0.303	0.303	0.007999 sec	0.029001 sec
k-NN (2-NN)	0.321	0.3	0.01 sec	0.047 sec
k-NN (3-NN)	0.309	0.296	0.007 sec	0.047001 sec
Decision Tree (max features: None)	0.294	0.292	0.044999 sec	0.001003 sec
Decision Tree (max features: sqrt)	0.282	0.282	0.006001 sec	0.001 sec
Decision Tree (max features: log2)	0.355	0.35	0.004 sec	0.0 sec
SVM	0.47	0.444	0.073998 sec	0.024 sec
Random Forests (num trees: 10, max features: sqrt)	0.436	0.422	0.035551 sec	0.001955 sec
Random Forests (num trees: 10, max features: log2)	0.415	0.398	0.027015 sec	0.001999 sec
Random Forests (num trees: 100, max features: sqrt)	0.433	0.413	0.322 sec	0.013001 sec
Random Forests (num trees: 100, max features: log2)	0.436	0.412	0.245999 sec	0.012001 sec

Mel-frequency cepstral coefficients (MFCCs)

Algorithm with parameters	Accuracy	F1	Training time	Testing time
Perceptron (alpha: 0.0001)	0.391	0.296	0.014001 sec	0.001 sec
Perceptron (alpha: 0.001)	0.391	0.296	0.013999 sec	0.001 sec
Perceptron (alpha: 0.01)	0.391	0.296	0.015 sec	0.0 sec
k-NN (1-NN)	0.358	0.364	0.008 sec	0.023001 sec
k-NN (2-NN)	0.352	0.349	0.012999 sec	0.02 sec
k-NN (3-NN)	0.348	0.353	0.006998 sec	0.016001 sec
Decision Tree (max features: None)	0.418	0.428	0.039998 sec	0.001001 sec
Decision Tree (max features: sqrt)	0.445	0.438	0.005999 sec	0.0 sec
Decision Tree (max features: log2)	0.379	0.381	0.003999 sec	0.001001 sec
SVM	0.712	0.707	0.077 sec	0.023999 sec
Random Forests (num trees: 10, max features: sqrt)	0.615	0.598	0.048999 sec	0.002 sec
Random Forests (num trees: 10, max features: log2)	0.558	0.546	0.031999 sec	0.002001 sec
Random Forests (num trees: 100, max features: sqrt)	0.7	0.684	0.380085 sec	0.013 sec
Random Forests (num trees: 100, max features: log2)	0.691	0.676	0.275002 sec	0.012 sec

Interpretation

Accuracy and F1 measure

As expected, the choice of the extracted feature used to train the models with the songs of the music data set had a huge impact on the prediction results.

When using the beats per minute (bpm) and the beats per minutes statistics (bpm_s) the classification accuracy was very poor. The F1 measure for bmp ranged from 0.012 (perceptron) to 0.134 (decision tree). The F1 measure for bmp_s ranged from 0.014 (perceptron) to 0.216 (random forests). The F1 measure was significantly higher when using the chroma feature extraction, ranging from 0.275 (perceptron) to 0.444 (SVM). The most accurate classification with an F1 measure of 0.707 was achived by using the Mel-frequency cepstral coefficients (MFCCs) in combination with SVM. The next accurate results were reached by random forests (up to an F1 of 0.684), the decison tree (up to an F1 of 0.438) and the k-NN (up to an F1 of 0.364). The least accurate results for the MFCCs was reached by the perceptron with an F1 measure of 0.296.

Training and testing time

The training time for bpm and bpm_s was quite similar and significantly faster than for the Chrome and the MFCCs (which were also quite similar). When inspecting the MFCC training times, it can be seen that the decision tree with the log2 limit function required the least time (0.004 seconds) and the random forests with 100 trees and the sqrt limit function required the most time (0.38 seconds).

The testing time was very similar among all extracted features with the perceptron being the fastest algorithm.

Confusion Matrix

	blues	**classical**	**country**	**disco**	**hiphop**	**jazz**	**metal**	**pop**	**reggae**	**rock**
blues	0.743	0.000	0.086	0.057	0.029	0.029	0.057	0.000	0.000	0.000
classical	0.000	0.970	0.000	0.000	0.000	0.030	0.000	0.000	0.000	0.000
country	0.000	0.000	0.788	0.030	0.000	0.030	0.030	0.030	0.030	0.061
disco	0.000	0.000	0.033	0.667	0.167	0.000	0.033	0.067	0.033	0.000
hiphop	0.000	0.000	0.037	0.000	0.741	0.000	0.037	0.037	0.148	0.000
jazz	0.000	0.000	0.094	0.000	0.000	0.844	0.000	0.000	0.031	0.031
metal	0.029	0.000	0.029	0.086	0.000	0.000	0.829	0.000	0.000	0.029
pop	0.000	0.000	0.030	0.152	0.061	0.000	0.000	0.697	0.030	0.030
reggae	0.034	0.034	0.103	0.069	0.069	0.000	0.000	0.034	0.655	0.000
rock	0.023	0.000	0.163	0.233	0.047	0.047	0.093	0.023	0.070	0.302

(columns: actual classes, rows: predicted classes)

When inspecting the confusion matrix of the best overall result (SVM with MCFFs) it can be seen that 97% of the tested classical songs were categorized correctly. More specifically, only 3% of the classical songs were classified as jazz songs. All other classical songs were correctly classified as classical songs. The genre the model had the most problems was rock. Only 30.2% of the rock songs were corretly classified. It seems to be most difficult for the model to distinct rock songs from disco songs. As many as 23% of all rock songs were classified as disco songs.

Comparison between data sets

For the small data set the decision tree delivered the most accurate results, for the big data set the random forests algorithm performed best. For the music classification the support vector machine performed best.