Technical Report

Informatik 2, SS2021
Parkhomenko Taisiya, 01650051

1 Introduction

The aim of my work was to practice object-oriented programming and gain an experience with supervised machine learning. Two following datasets were used:

Parkinson Speech Dataset This dataset contains data from speech recordings of subjects with Parkinson's disease as well as healthy subjects, it consist of 31 columns, one of which indicates the presence (or absence) of current disease in the patient.

Heart Disease Dataset This dataset contains heart disease diagnostics, it consist of 14 columns, one of which indicates the presence (or absence) of current disease in the patient.

More information for datasets on:

https://archive.ics.uci.edu/ml/datasets/heart+disease

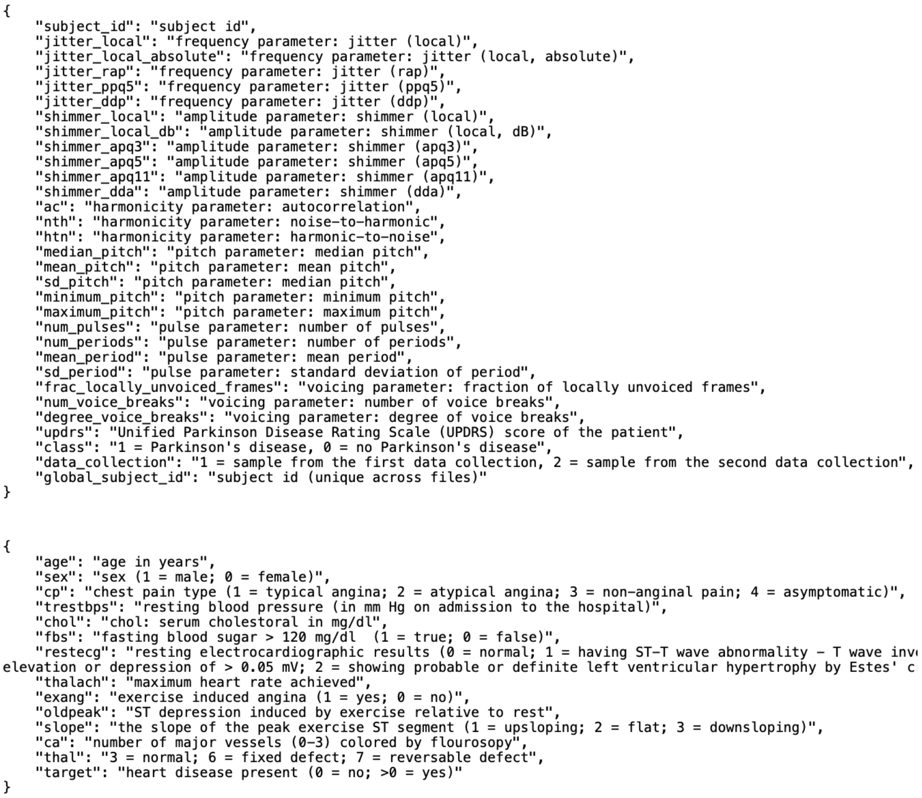
https://archive.ics.uci.edu/ml/datasets/Parkinson+Speech+Dataset+with++Multiple+Types+of+Sound+Recordings

2 Experimental Setup

Setup and comparison of classifiers Three classifier were imported with *Sklearn* - machine learning library for the Python programming language, last classifier was self-implemented:

- **Dummy Classifier** - the most frequent label in the training set will be predicted.
- **Support Vector Classifier**
- **Logistic Regression Classifier** - Implements regularized logistic regression
- **kNN** - The input consists of the k closest training examples in data set. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.

Used features I have used all features (except ”target”) from feature dictionaries (Figure 1) provided by *.json* files included in both datasets.



```
{
    "subject_id": "subject id",
    "jitter_local": "frequency parameter: jitter (local)",
    "jitter_local_absolute": "frequency parameter: jitter (local, absolute)",
    "jitter_rap": "frequency parameter: jitter (rap)",
    "jitter_ppq5": "frequency parameter: jitter (ppq5)",
    "jitter_ddp": "frequency parameter: jitter (ddp)",
    "shimmer_local": "amplitude parameter: shimmer (local)",
    "shimmer_local_db": "amplitude parameter: shimmer (local, dB)",
    "shimmer_apq3": "amplitude parameter: shimmer (apq3)",
    "shimmer_apq5": "amplitude parameter: shimmer (apq5)",
    "shimmer_apq11": "amplitude parameter: shimmer (apq11)",
    "shimmer_dda": "amplitude parameter: shimmer (dda)",
    "ac": "harmonicity parameter: autocorrelation",
    "nth": "harmonicity parameter: noise-to-harmonic",
    "htn": "harmonicity parameter: harmonic-to-noise",
    "median_pitch": "pitch parameter: median pitch",
    "mean_pitch": "pitch parameter: mean pitch",
    "sd_pitch": "pitch parameter: median pitch",
    "minimum_pitch": "pitch parameter: minimum pitch",
    "maximum_pitch": "pitch parameter: maximum pitch",
    "num_pulses": "pulse parameter: number of pulses",
    "num_periods": "pulse parameter: number of periods",
    "mean_period": "pulse parameter: mean period",
    "sd_period": "pulse parameter: standard deviation of period",
    "frac_locally_unvoiced_frames": "voicing parameter: fraction of locally unvoiced frames",
    "num_voice_breaks": "voicing parameter: number of voice breaks",
    "degree_voice_breaks": "voicing parameter: degree of voice breaks",
    "updrs": "Unified Parkinson Disease Rating Scale (UPDRS) score of the patient",
    "class": "1 = Parkinson's disease, 0 = no Parkinson's disease",
    "data_collection": "1 = sample from the first data collection, 2 = sample from the second data collection",
    "global_subject_id": "subject id (unique across files)"
}
```

```
{
    "age": "age in years",
    "sex": "sex (1 = male; 0 = female)",
    "cp": "chest pain type (1 = typical angina; 2 = atypical angina; 3 = non-anginal pain; 4 = asymptomatic)",
    "trestbps": "resting blood pressure (in mm Hg on admission to the hospital)",
    "chol": "chol: serum cholestoral in mg/dl",
    "fbs": "fasting blood sugar > 120 mg/dl  (1 = true; 0 = false)",
    "restecg": "resting electrocardiographic results (0 = normal; 1 = having ST-T wave abnormality - T wave inv
elevation or depression of > 0.05 mV; 2 = showing probable or definite left ventricular hypertrophy by Estes' c
    "thalach": "maximum heart rate achieved",
    "exang": "exercise induced angina (1 = yes; 0 = no)",
    "oldpeak": "ST depression induced by exercise relative to rest",
    "slope": "the slope of the peak exercise ST segment (1 = upsloping; 2 = flat; 3 = downsloping)",
    "ca": "number of major vessels (0-3) colored by flourosopy",
    "thal": "3 = normal; 6 = fixed defect; 7 = reversable defect",
    "target": "heart disease present (0 = no; >0 = yes)"
}
```

Figure 1: Feature dictionaries for Parkinson and Heart disease datasets.

Size of the test Parkinson Speech Dataset `"test_size": 0.33`, Number of Instances: 1040, Number of Attributes: 26.

Heart Disease Dataset `"test_size": 0.33`, Number of Instances: 303, Number of Attributes: 75.

Metrics for evaluation Classifiers, their accuracy and confusion matrices.

Handling of missing values Missing values are specified with *n/a* for the Parkinson Speech Dataset and *?* for the Heart Disease Dataset the data files.

In Python code missing values were read from config files:

`config_parkinson_sound_recording.json` and `config_heart_disease.json`.

Since they are not the same and can not be understood by Python, a non-public variable was created:

```
self._na_characters = config_info_dict['na_characters']
```

Then, the variable of missing values was passed into Data Frame:

```
self._data.to_csv(output_file, index=False, index_label=False, ...
                                ...na_rep=self._na_characters[0])
```

3 Results and Discussion

A *baseline* is the result of a very basic model or solution. After creating a baseline, try to make more complex solutions in order to get a better result. It is good, if a better score achieved than the baseline. (stackexchange.com)

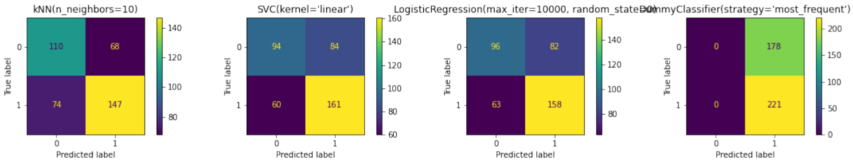
On the basis of Parkinson Disease Dataset Figure 2 we see, that Self-implemented kNN Classifier, Support Vector Classification and Logistic Regression Classifier performed better than baseline - Dummy Classifier.

On the basis of Heart Disease Dataset Figure 3 we see, that Support Vector Classification and Logistic Regression Classifier performed best. Self-implemented kNN classifier less efficient, but better than baseline - Dummy Classifier.

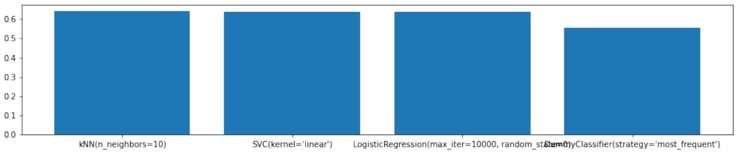
4 Conclusion

Despite the fact that the Accuracy Scores of best classifiers are equal, it can be seen on Figure 2, 3 (a) and (b), that the Confusion matrices are not the same. Meaning that the predicted data differ from classifier to classifier. Generalizing following facts, we can not say which of the classifiers with equal Accuracy Scores is the best, it need to be decided according to the situation and the task.

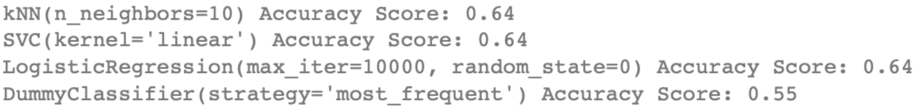
Further improvements of the prediction Based on the results I have obtained in this assignment, to improve results and discussion, I would plot Accuracy Scores of kNN classifier over different k (neighbor number). To see at which k-value, classifier predicts best, since bigger k smooth the effect of data-outliers.



(a) Confusion matrices for each evaluated classifier.



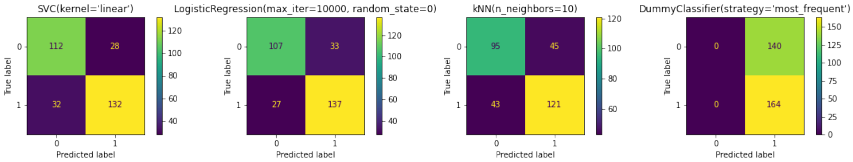
(b) Accuracy of all evaluated classifiers.



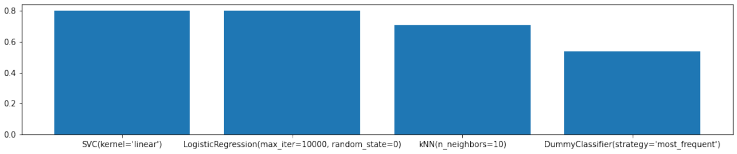
```
kNN(n_neighbors=10) Accuracy Score: 0.64
SVC(kernel='linear') Accuracy Score: 0.64
LogisticRegression(max_iter=10000, random_state=0) Accuracy Score: 0.64
DummyClassifier(strategy='most_frequent') Accuracy Score: 0.55
```

(c) Classifiers ranked by accuracy.

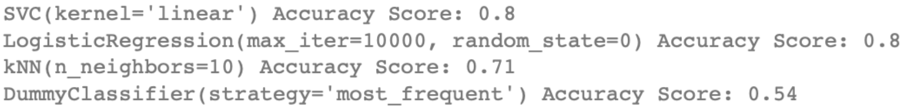
Figure 2: Visualized classification results for Parkinson Disease Dataset.



(a) Confusion matrices for each evaluated classifier.



(b) Accuracy of all evaluated classifiers.



```
SVC(kernel='linear') Accuracy Score: 0.8
LogisticRegression(max_iter=10000, random_state=0) Accuracy Score: 0.8
kNN(n_neighbors=10) Accuracy Score: 0.71
DummyClassifier(strategy='most_frequent') Accuracy Score: 0.54
```

(c) Classifiers ranked by accuracy.

Figure 3: Visualized classification results for Heart Disease Dataset.