Project in artificial intelligenceTO: Prof. Shaul Markovitch

A picture containing light, sitting, green, blue

Description automatically generated

Emil Khshiboun

TAble of  
Contents

Project DESCRIPTION 1

Implemention 2

Decision Tree 2

K-Nearest Neighbor 4

datasets 5

Fifa19 5

Authentic Bank Notes 6

Breast Cancer 6

Wine Quality 7

EXPERIMENTS 8

Fifa19 experiments 9

Breast Cancer experiments 10

Wine Quality 11

Authentic Banknotes 12

Conclusions 13

source code 14

git repository 14

running 14

# Project DESCRIPTION

Project goal is to implement four machine-learning algorithms and test their performance on four different datasets.

Algorithms implemented in the project are:

* **Decision Tree** – Parameters: m1, minimum number of samples in a leaf
* **Stochastic Decision Tree** – Parameters: m2, same as normal Decision Tree, the difference is the way we choose the next feature to split and internal node.
* **K-Nearest Neighbor** – Parameters: K number of neighbors to vote
* **Stochastic K-Nearest Neighbor** – Parameters: K, same as the normal KNN, the difference is in the decision phase, we pick K nearest neighbor **randomly** from the nearest 2\*k neighbors.

# Implemention

The whole projected was written in Python (version > 3.x).

Paths are relative to the project root directory

### Decision Tree

File name: ID3.py

Parameters:

* • min\_leaf\_samples - The minimum number of samples to split a node
* features - List of feature a class that contains the type of a feature (such as continuous,discrete)
* stochastic – bool to indicate if the tree is stochastic decision tree or not.

The training phase of this model accepts as input labeled samples and build a decision tree recursively, the build is stopped when no more features are available or the current node has less than min\_leaf\_samples samples, in this case, a leaf is created and its classification is decided by the majority of its samples.

The feature to split according to is decided by calculating the information gain of each feature (according to node’s entropy). If stochastic parameter is set, the model chooses a random feature with each feature given probability of being chosen proportional to its information gain, . if the stochastic parameter is not set, the feature with the maximum information gain is choosen.

Depending on the feature type, the feature might be removed from the features pool, this is decided by the feature type, a Discrete feature is removed from the features pool, if it’s continuous the feature is not removed.

At the end of the process, the tree is saved, the samples are discarded from the memeory.

Predicting samples is done by traveling the tree that was built in the training phase, according to the feature of the current node, the classification of a sample is the same as leaf node that it arrives to.

### K-Nearest Neighbor

File name: KNN.py

Parameters:

* K – Is the number of neighbors to look at when predicting a sample
* dist\_fun – A distance function to be used when comparing distance between two samples, the default is Euclidian distance.
* stochastic – Bool to indicate if the model is stochastic or not.

The training phase of this model accepts as input labeled samples, the training is as simple as saving the samples in the memory. That’s it.

Predicating samples is done by calculating distance between each sample to all the samples that were saved in the memory, the distance is calculated with the provided dist\_func parameter.

If the model stochastic, then from these distance, we choose 2\*K minimum samples, then from this list, we randomly pick k samples, and the decision is taken by these k samples, each sample ‘votes’ on the classification that should be given to sample we want to predicit.

The classification with the highest votes gets choosen.

If the model is not stochastic, then we choose K minimum samples (not 2\*K). the reset is the same as above.

# datasets

### Fifa19

Path: datasets/fifa19/

Source: <https://www.kaggle.com/karangadiya/fifa19>

This dataset includes attributes about football players as seen in FIFA database.

Attributes includes different rating aspects of the player such as performance on different part in the field, drippling abilities, defenece abilities and many other more..

The target field is “Overall rating”

Modifications:

In the original dataset, this field was a number between 0-100, this was reduced to 10 different rating scale, from 1 to 10. (0-10 -> 1, 90-100 -> 10)

Some columns were deleted such as player photo, club photo, and others.

### Authentic Bank Notes

Path: datasets/banknotes

Source: <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>

“Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images.”

Target field is 1 if the the banknote is authentic bank note, 0 otherwise.

### Breast Cancer

Path: datasets/cancer

Source: <https://www.kaggle.com/uciml/breast-cancer-wisconsin-data>

Detailed information about the dataset and the attributes are in the link.

The classification done here is to predicit if the cancer is malignant or benign.

### Wine Quality

Path: datasets/wine/

Source: https://archive.ics.uci.edu/ml/datasets/wine+quality

Samples related to red and white variants of the Portuguese "Vinho Verde" wine. More details about the attributes are in the source link.

Modifications:

The target field “Quality” had 11 classifications from 0-10, I’ve changed it 0,1 classification problem, where 1 means high quality wine, 0 is low quality wine.

# EXPERIMENTS

Experiments descriptions:

Experiments [1-4] were run on forest trees, where each forest had n different tree of the same classifier where n = 1,3, 5…, 21

All experiments were run through 10-CrossValidation to find the optimum model parameter (k in KNN, min\_leaf\_samples in ID3)

The parameter that is choosen is then used by the Forest Classifier, each tree of the n trees uses the same parameter.

Features were normalized using MinMax algorithm on KNN & stochastic KNN models.

Each tree used of the train samples randomly.

Experiment 5 was run on forest tree were each forest had n trees, where n = 4,8,12,16,20 trees, the each forest had 4 classifiers, Decision Tree, Stochastic Decision Tree, KNN, Stochastic KNN, the number of classifier of each classifier is n/4.

### Fifa19 experiments

Chart

Description automatically generated

### Breast Cancer experiments

Chart, line chart

Description automatically generated

### Wine Quality

Chart, line chart

Description automatically generated

### Authentic Banknotes

Chart, line chart

Description automatically generated

All the logs and the graphs (including more scaled graphs for each experiment per dataset) can be found under “results” directory.

# Conclusions

As we can see from the graph, Decision Tree, Stochastic Decision Tree and KNN performed well on each dataset, however the stochastic KNN had much lower accuracies.

Like we described earlier, stochastic KNN chooses **randomly** K samples from the 2\*k nearest samples, this approach messed the whole concept of the KNN, as the model gives more distant samples the ability to vote.

One might ask why the stochastic decision tree performed just as well as the normal decision tree. The answer is simple, stochastic decision trees choose the next feature randomly with distribution that is proportional to its information gain, that means features with higher information gain are more likely to be chosen. Moreover, the datasets we used had continuous features which means when a feature is used, it’s not removed from the features pool, and might be chosen again. Therefore, picking a “bad” feature at some point because of the “randomness”, does not impact the whole tree in the long run.

We conclude as well, that the more trees we have in the forest, the higher the accuracy is, which makes the use of forest trees a better choice.

# source code

## git repository

Source code of the project can be found under my git repository @ <https://github.com/Emilyos/AIProject>

## running

Detailed information is shown when running the the main with -h command:

python main.py -h

“usage: main.py [-h] --exp EXP --dataset DATASET [--kFold KFOLD] [-p P [P ...]] [-K K] [-M M]”

For example running experiment 1 on “Wine Quality” dataset with 10-fold Cross validation with parameters to check (in this case the parameter is min\_leaf\_samples) = [20,30,50,100]

python main.py --exp 1 --dataset wine --kFold 10 -p 20 30 50 100

the output of the program (including the graph that is created) in saved in the “results” directory.