MACHINE LEARNING

SHEET 3

16.12.2021

11:00 A.M. UNTIL 12:00 A.M. VIA ZOOM

Please prepare the exercises in order to present them in the meeting. Fill-in the exercises you solved in the questionnaire in StudIP until 15.12.2021 4:00 p.m.
Join the meeting via this Zoom link:
https://uni-trier.zoom.us/j/81020909373?pwd=aHFsVjQ0TVBGbHlIVFJlZWt3THJoQT09

TASK 1: BAGGING K-MEANS

In this exercise, we work with the Wisconsin Breast Cancer Dataset (https://scikit-learn.org/stable/datasets/index.html#breast-cancer-dataset). We will look at the possibilities that scikit-learn provides for using ensembles of ML models.

a) Fetch the Wisconsin Breast Cancer dataset with the respective scikit-learn methods and split into training and test set (test ratio of 0.2).
b) Train a k-Means clustering algorithm ($k = 2$) on the training data and inspect its performance on the test set w.r.t. the Adjusted Rand Index (ARI). What does this measure express?
c) Train a BaggingClassifier with k-Means ($k = 2$) as the base estimator. The classifier should use all features and 30 % of the training examples for each created classifier. Also, it should be allowed to use the same example for different classifier instances. There should be 20 estimators used during bagging.
d) Evaluate the ARI performance of the bagging classifier. How does it perform? In which scenarios is bagging probably outperforming a single estimator?

TASK 2: RANDOM FOREST

Random forests combine the predictions of multiple Decision Trees. They also provide insight on the importance of features.

a) Generate some synthetic regression data with the respective scikit-learn method (https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_regression.html#sklearn.datasets.make_regression). You want to have 1000 samples with 20 features (10 informative ones). Set a standard deviation for the gaussian noise of 0.2.
b) Build a pipeline that scales the data and afterwards applies a Random Forest regressor. Set the parameters of the Random Forest to 50 estimators and a maximum number of ten leaf nodes. How does a Random Forest estimator work?
c) Train this pipeline and measure its performance on the test set. Use a suitable performance measure.
d) Check the importance values of the features estimated by the Random Forest. Are they compliant with the generated data?

DEPARTMENT OF BUSINESS INFORMATION SYSTEMS II
PROF. DR. RALPH BERGMANN
UNIVERSITY OF TRIER

Task 3: AdaBoost with Logistic Regression

Boosting of classifiers is besides bagging another method of creating ensemble learners that are supposed to outperform standalone estimators. We will use a Logistic Regression classifier in a boosting setup on the Wisconsin Breast Cancer dataset.

a) Fetch the Wisconsin Breast Cancer dataset with the respective scikit-learn methods and split into training and test set (test ratio of 0.2). Scale the data to [0,1].
b) Train a Logistic Regression that classifies the training data. Use default parameters. What is the performance of the trained model w.r.t. accuracy?
c) Train an AdaBoostClassifier with Logistic Regression (same configuration as before) as the base estimator. There should be 20 estimators used during boosting.
d) Compare the accuracy on the test set of the boosted approach and the standard approach. How does AdaBoost work?

Task 4: Principal Component Analysis For Dimensionality Reduction

Principal Component Analysis (PCA) is an approach to reduce the dimensionality of a dataset with the goal of preserving as much variance as possible. We will perform a PCA on a synthetic dataset.

a) Create a synthetic regression dataset with ten toy examples. Use the respective scikit-learn function for this (https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_regression.html#sklearn.datasets.make_regression). The data should be 2D with a single scalar as the target (so 3D for the complete dataset). Center the data by subtracting the mean value from all data points.
b) Visualize the data with matplotlib in a 3D plot where the data points are on the X- and Y-axis and the target values are on the Z-axis.
c) Perform a linear regression with the data and visualize the linear model together with the data in a 3D plot. You can simply add the linear model to the visualization from part b) of this exercise.
d) Compute the Principal Components (PCs) of the data using the Singular Value Decomposition (SVD) of numpy. Project all data points onto the prior computed axis resulting from SVD.
e) Visualize the resulting data and the respective targets in a 2D scatter plot. Compare it with the visualization of the data transformed by the PCA estimator that is provided by scikit-learn.

Optional Task 5: Principal Component Analysis For Anomaly Detection

Principal Component Analysis (PCA) is an approach to reduce the dimensionality of a dataset with the goal of preserving as much variance as possible. It can also be used for anomaly detection by analyzing the errors when transforming from a lower-dimensional representation to the original representation.

a) Load the Iris dataset (https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html) with the respective scikit-learn method.
b) Perform a PCA (2 resulting dimensions) with the respective scikit-learn method and visualize the transformed dataset in a scatter plot. Make sure that the datapoints are colored according to their class.
c) Compute the reconstruction error of the transformed datapoints by executing the following steps:
    1. Recreate the original dimensionality by performing an inverse transform with the learned PCA.
    2. Compute the Mean Squared Error (MSE) between every example from the inverse transformed data and the original data. The mean should be computed across the feature errors of every example.
d) Visualize the error of every example in a plot. Sort the values beforehand. Color the datapoints according to their class. Interpret the resulting plot.

Hints for Optional Tasks

Task 5 on this sheet is an optional task. This means that you can solve this exercise if you want, but you do not have to do it. For the percentage calculation this means, that the point that you collect with such a task is added to your overall points of solved tasks but not to the complete number of tasks to solve. Thus, a person who solves all tasks including optional ones can collect more than 100% of the points. Please check each task on upcoming sheets if it is an optional or a regular one.