Artificial Intelligence 2022/2023

Exercise Sheet 5: Supervised Learning

5.1 Software/Library Installation

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including neural networks, support vector machines, random forests, gradient boosting, k-means and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Start by installing Python, Anaconda, Jupyter Labs, NumPy, SciPy, Pandas, Scikit-Learn, MatPlotLib and Seaborn. In fact, it is only needed to install Anaconda that contain all the others following the link: https://www.anaconda.com/products/individual

Information about the rest of the packages/libraries may be found at:

- Python Website, https://www.python.org/
- Anaconda Website, https://www.anaconda.com/
- Project Jupyter Website, https://jupyter.org/
- NumPy Website, https://numpy.org/
- SciPy Website, https://www.scipy.org/
- Pandas Website, https://pandas.pydata.org/
- Scikit-Learn Website, https://scikit-learn.org/
- MatPlotLib Website, https://matplotlib.org/
- Seaborn Website, https://seaborn.pydata.org/

After installing all the libraries, please open the example Notebook available at moodle that contains an example code containing several exercises.

5.2 Iris flower data set – Data Preprocessing and Simple Classification

From Wikipedia - The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper “The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis”. It is sometimes called Anderson's Iris data set because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species. Two of the three species were collected in the Gaspé Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus".

The data set consists of 50 samples from each of three species of Iris (Iris Setosa, Iris Virginica and Iris Versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other. In this exercise we will use simple machine learning algorithms to analyze the dataset and create a model to classify the flowers in their specific type.

a) Unzip the file with the example notebook available at moodle.

b) Import the Pandas library, read the data from the CSV file and check the data using the head(), describe(), and other Pandas commands.

c) Read again the data identifying as missing the values marked with 'NA'.

d) Import the MatPlotLib and Seaborn libraries and create a scatterplot matrix of the data.

e) After looking at the plot it seems that the field researchers make some errors inserting the data. It sounds like one of them forgot to add Iris- before their Iris-versicolor entries. The other extraneous class, Iris-setossa, was simply a typo that they forgot to fix. Use the DataFrame to fix these errors. Create a new scatterplot of the data.

f) Looking at the scatter plot, since it is impossible to have any 'Iris-setosa' rows with a sepal width less than 2.5 cm, drop those values and create an histogram with the 'Iris-setosa' sepal width.

g) The next data issue to address is the several near-zero sepal lengths for the Iris-versicolor rows. Those rows were gathered in meters instead of cm. Please correct that mistake and draw the corresponding histogram.

h) One way to deal with missing data is mean imputation. Do that for the missing values of the petal widths for Iris-setosa and create a new scatter plot for the data.

i) Save the new clean dataset to the disk with the name "iris-data-clean.csv".

j) Create some violin plots of the data to compare the measurement distributions of the classes. Violin plots contain the same information as box plots, but also scales the box according to the density of the data.

k) Create two variables with the inputs and labels using the clean dataset created.

l) import the train_test_split and create randomly training and testing sets with 75% of the examples on the training set and 25% on the testing set: training_inputs, testing_inputs, training_classes, testing_classes

m) Import the DecisionTreeClassifier and train the classifier on the training set showing the final score/accuracy.

n) Experiment 1000 times the classifier and plot a histogram of the obtained accuracies.

o) Import StratifiedKFold and use stratified cross-validation with 10 splits and train again the data.

p) Import GridSearchCV and perform a Grid Search over the Decision Tree parameters to find the best parameters, visualizing the grid with the accuracies for each parameter's pairs (max_features 1-4 and max_depth 1-5).

q) Visualize in a graphical manner the final decision tree achieved.