
Problem assignment 1

Due: Thursday, January 27, 2022

This homework assignment will ask you to implement and submit short python programs to analyze, preprocess and visualize data. The basic packages we will rely on in this course are:

- Numpy: arrays, matrices, numerical computation tools; https://numpy.org/
- Pandas: data, loading and saving files in different format https://pandas.pydata.org
- Scipy: algorithms for scientific computing; https://scipy.org)
- Matplotlib: data visualization, graphs; https://matplotlib.org
- Scikit-learn: machine learning libraries and models; https://scikit-learn.org/stable/

Problem 1. Matrix operations

Let us assume:

$$v = \begin{bmatrix} 9 & 5 & 10 \end{bmatrix}$$

$$u = \begin{bmatrix} 4 \\ 1 \\ 3 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$$

$$B = \begin{bmatrix} 7 & 1 & 9 \\ 2 & 2 & 3 \\ 4 & 8 & 6 \end{bmatrix}$$

$$C = \begin{bmatrix} 8 & 6 & 5 \\ 1 & -3 & 4 \\ -1 & -2 & 4 \end{bmatrix}$$

Please calculate (you may use Python's Numpy library to accomplish this task):

- $u^T * u$
- $u * u^T$
- $v * u$
- $u + 5$
- A^T
- $B * u$
- B^{-1}
- $B + C$
- $B - C$
- $A * B$
- $B * C$
- $B * A$

Report the results.

Problem 2. Exploratory data analysis

In this Problems 2-4 of this assignment we will explore and analyze the Pima dataset. The dataset consists of 8 attributes and a binary attribute defining the class label, the presence of diabetes. Data entries are organized in rows such that attributes come first and the class label is last. The description of the dataset can be found in *pima_desc.txt* file.

You are given a python file $hw - 1 - pima.py$ that loads the pima dataset and splits it to X and Y components. You will gradually add the code to this file to generate the answers for the remaining questions in the assignment. You will submit the modified file as $hw - 1 - pima - modified.py$. Please note the answers, figures, and the related analysis answering the specific questions should be reported in the Report.

Answer the following questions with the help of Python:

- (a) What is the range (minimum and maximum value) for each of the attributes? Hint: use Python's Numpy functions *min* and *max*.
- (b) What is the mean and standard deviation of each attribute? Hint: Use Python's Numpy *mean* and *std*.
- (c) Split *pima.txt* data into two data subsets - one that includes only examples with class label "0", the other one with class "1" values. Calculate and report the mean and standard deviations of each attribute (columns 1-8) on these two subsets. Analyze the means and standard deviations of attribute values and select the attribute you think should be most helpful in discriminating the two classes. Include the attribute name in the report and explain why you think the attribute is the best for discriminating the two classes.
- (d) Calculate and report correlations between the first 8 attributes (in columns 1-8) and the target class attribute (column 9). Use Python's corrcoef function to do the calculations. What is the attribute with the highest (positive) correlation to the target attribute? Do you think it is the most or the least helpful attribute in predicting the target class? Explain.
- (e) Calculate all correlations between 8 attributes (using Python's corrcoef function). Which two attributes have the largest mutual correlation in the dataset?

While the analysis using basic statistics as performed above conveys a lot of information about the data and lets us make some conclusions about the importance of attributes for prediction or their mutual relation, it is often very useful to inspect the data also visually and get more insight into various shapes and patterns they hide. In the following we will inspect the data using histograms and 2D scatter plots.

- (f) **Histogram analysis** gives us more information about the distribution of attribute values. You can generate histogram plots using matplotlib.pyplot's function hist. Analyze all attributes in the data using the hist function and 20 bins. Answer the following questions: Which histogram resembles most the normal distribution? In your report show at least two histograms, including the choice you picked as the most normally distributed attribute.
- (g) Histogram analysis in part (f) lets you plot the distribution of values for any input data. We we can use hist functions also to look at attribute distributions for class 1 and class 0 individually and compare them. Similarly to part (c) divide pima dataset into two datasets, one with instances corresponding to class 0 and the other one corresponding to class 1. For each attribute in columns 1 and 8 plot two histograms of the attribute values, one for class 1 and the other one for class 0. Compare the two histograms for each attribute. Based on the pairs of histograms choose an attribute you think should be most helpful in discriminating the two classes.

Include the attribute name and the histograms for that attribute for class 1 and class 0 in the report. Explain why you think the attribute is the best.

- (h) **2D Scatter plots** plots let us inspect the relations between pairs of attributes. Write (and submit) a function *scatter_plot* that takes pairs of values for two attributes and plots them as points in 2D (use matplotlib.pyplot's function scatter to do the plot). Analyze the pairwise relations between 8 attributes in the pima dataset using the scatter plot function. Answer the following questions. What scatter plot would you expect to see for the two dimensional space if the two attributes are independent and random? Do you see any interesting non-random patterns among the pairs? Include two scatter graphs you think show some interesting dependences or patters. Explain why you think these are interesting? Do not forget to include with every plot the corresponding attribute names.

Problem 3. Data preprocessing

Before applying learning algorithms some data preprocessing may be necessary. In this problem we explore three preprocessing methods: transformation of categorical values to (safe) numerical representation, normalization of continuous values, and discretization of continuous values.

- (a) Assume you have an attribute with 8 categorical values {brown, blue, white, red, yellow, orange, green, black}. Devise one-hot encoding of the values and explain in the report how values are mapped. Use the mappings to convert the following vector of attribute values to one hot representation and include the results in the report:

$$\begin{bmatrix} red \\ black \\ yellow \\ red \\ green \\ blue \\ blue \end{bmatrix}$$

- (b) One important preprocessing step often applied to datasets is to normalize the data attributes so that they are on approximately the same scale. One classic approach normalizes the data attribute values according to their mean and standard deviation. Briefly, to calculate the normalized value we apply following formula:

$$x_{\text{norm}} = \frac{x - \mu_x}{\sigma_x}.$$

where x is an unnormalized value, μ_x is the mean value of the attribute in the data and σ_x its standard deviation. This normalization method is implemented in the

sklearn.preprocessing library and is named StandardScaler. Please use the method on the X data of pima. Report new normalized values of the attribute 3 for the first five entries of the pima data.

- (c) Another preprocessing step commonly applied to data is related to discretization of real-valued attributes to K equal sized bins. This discretization method is implemented in the sklearn.preprocessing library and is called KBinsDiscretizer. Use this function and apply it to attribute 3 of the pima dataset by assumimg 10 equal size bins. Report new (discretized) values of the attribute 3 for the first five entries in the pima dataset.

Problem 4. Splitting data into training and testing sets

In this problem we explore Python's functions supporting the splitting of the dataset into the training and testing sets. To do so we will use function *train_test_split* that is implemented in sklearn.modelselection library.

- (a) Run the *train_test_split* on the pima data with test size set to 0.33, and random seed set to 7. Calculate and report the size of the training and testing dataset.
- (b) Run the *train_test_split* on the pima data with the test size set to 0.33, and the random seed set to 3. Calculate and report the size of the training and testing dataset. Compare the splits generated in part a and part b. Are the training and testing data the same or different in terms of instances.
- (c) Run the *train_test_split* on the pima data with test size set to 0.25, and random seed set to 7. Calculate and report the size of the training and testing dataset.