Amirkabir University of Technology
(Tehran Polytechnic)

Machine Learning Course By Dr. Nazerfard

CE5501 | Fall 2023

Teaching Assistants

Atiyeh Moghadam

Zahra Akhlaghi

Zahra Zanjani

Assignment (2)

Outlines. In this assignment, K-NN and Decision Tree are noticed as well as evaluation metrics.

Deadline. Please submit your answers before the end of November 13th in courses.aut.ac.ir. Other methods like sending via email or in social networks are not accepted and will not be considered.

Assignment Manual

Delay policy. During the semester, you have extra 10 days for submitting your answers with delay. Mentioned time is for all assignments. After that, for each day of delay you loss 20% points of that assignment. After 4 days you miss all points and any submit doesn`t acceptable. Remember that saving this time doesn`t have any extra point.

Sharing is not caring. Students are free to discuss and share their ideas about problems with others. But sharing source codes, solutions, answers and other results is not allowed and based on university`s rule, both sides will be graded zero.

Problems are waiting you. Some problems are required to be implemented within a programming language and obtain some charts, images, results, etc; then discuss about it. These types of questions are tagged by #Implementation. Some other problems are required to be solved or computed by hand or research about them. These types of questions are tagged by #Theorical. You are not allowed to use programming language or other technical tools to answer theorical problems.

Report is the key. All students' explanations, solutions, results, discuss and answers must be compacted into a single pdf report. A clean and explicit report is expected and may followed by extra pts; so, you may need to write any related detail or experience during the solving problems. Report file should started within a cover page that it includes course and assignment information as well as identical details like name, student number and email address. Second page should be table of contents that indicates student`s answer to each question. Please repeat your name and student number in left side of footer in other pages. Also, you are free to write in Persian or English. If typing is bothering you, so write in a paper and put its picture with acceptable readability quality in report file.

Organize the upload items. Students should upload their implementation source codes as well as results and report. You should upload a single .zip file with the following structure:

ML_01_[std-number].zip
- Report
  - ML_01_[std-number].pdf
  - [other material and results]
- Source codes
  - P[problem-number]_[a-z].py
  - P[problem-number]_[a-z].ipynb
  - …

Python is the power. Students are free to use any programming language like python, matlab, C++ , etc. However it is recommended strongly to use python in jupyter notebook environment; so, you may need to upload your .py or .ipynb sources.

Feel free to contact. If you have any question or suggestion, need guide or any comment be comfortable to ask via email as well as Telegram group.

Problem 1: Why and how (15 pts)

#Theorical

a) Is the Gini Impurity of a node in the decision tree lower or greater than that of its parent? Comment whether it is generally lower/greater or always lower/greater.

b) What is the space and time complexity of the k-NN Algorithm?

c) Is feature scaling required for K-NN? Describe the methods used for feature scaling in k-NN algorithm?

d) What is the space and time complexity of the k-NN Algorithm?

e) Why is it recommended not to use the k-NN Algorithm for large datasets? How does the curse of dimensionality affect KNN algorithm?

f) Use k-nearest neighbor classification to classify on the following training set with two classes {+, -}:

class	x1	X2
+	1	2
+	2	1
-	2	2
+	2	3
-	3	1
+	3	2

(i) Plot these six training points on an X1-X2 axis.

(ii) Separate the classes {+, -} with a decision boundary on your plot in part a for k = 1. Fill in the region(s) that will classify the "_" classes.

(iii) Separate the classes {+, -} with a decision boundary on your plot in part a for k = 3. Fill in the region(s) that will classify the "-" classes.

g) Can decision trees handle non-linear relationships between features and the target variable? Then how can we approximate non-linear relationships using decision trees?

h) What is pruning in the context of decision tree? What types of pruning do we have?

i) Show the decision tree created by the ID3 algorithm for spam detection using the given training instances. Include the information gain calculations, class values, and associated instances for each leaf node.

No.	attention	congrats	money	Spam/non-spam
1	1	0	0	Spam
2	0	0	1	Spam
3	0	0	0	Non-spam
4	1	1	0	Non-spam
5	0	0	0	Non-spam

6	1	0	1	Spam
7	0	1	1	Non-spam
8	1	0	0	Spam
9	0	0	0	Non-spam
10	1	0	0	Spam

Problem 2: Stroke Prediction(35 pts)

#Implementation

According to the World Health Organization (WHO), stroke is the second leading cause of death worldwide, responsible for approximately 11% of all fatalities. In this assignment, you will use machine learning to classify whether a person will have a stroke, considering the imbalanced nature of the data.

I. Instructions

a) Load the stroke.csv dataset and describe it using a correlation matrix.
b) Perform the necessary preprocessing steps, such as handling missing values and normalization. Explain the preprocessing steps in your document.
c) Bias in datasets can lead to inaccurate predictions. Identify any potential sources of bias in the stroke dataset.
d) Use a stratified sampling technique to split the dataset into training, validation, and test sets, ensuring that the target variable is balanced in all three sets.
e) Train a KNN (K-Nearest Neighbors) model to classify whether a person is at risk of having a stroke. Utilize the elbow method to tune the hyperparameters of the model and optimize its performance on the validation set.
f) Train a decision tree model to classify whether a person will have a stroke or not.
g) Tune the hyperparameters of the decision tree model using grid search to optimize its performance on the validation set.
h) Compare the performance of the two models and select the best performing model.
i) Explain Which metric is more important in this problem, precision or recall? (why?)
j) Evaluate the performance of the best performing model on the test set using the accuracy and F1-score metrics, both in macro and micro stages.
k) Discuss the different ways to deal with imbalanced data in classification problems. Apply one of these methods to improve the performance of the best performing model.

II. Allowed Libraries

Feel free to utilize your preferred machine learning library to complete this assignment.

III. How to report the results:

you should include the following information in the results of your experiments.

- The name of the model
- The cross-validation results (mean and standard deviation of the accuracy, precision, recall precision, recall and F1-score metrics, both in macro and micro stages)
- The performance of the best performing model on the test set (accuracy, precision, recall and F1-score metrics)

Problem 3: KNN Is More!(25 pts)

#Implementation

Assume you have started working for an insurance company that has amassed a vast amount of information from its customers. Your initial task is to discern which of this information is informative. In the classroom, you encountered K-Nearest Neighbors (KNN) for classification. However, K-Nearest Neighbors can provide us with even more insights about the data. In this problem, we will employ KNNs to learn weights for the available features and subsequently select the features with the highest weights.

Figure 1: What stable diffusion thinks of you as a data scientist in an insurance company who has a lot of frustration finding good features.

a) Load the dataset. What are the necessary preprocessing steps for a KNN problem? Perform them if needed. Then, split the dataset into training and test sets with a ratio of 0.9:0.1.

b) **Implement** a function to calculate the loss of weighted KNN using the following steps:

1. For all observations in the dataset, calculate the weighted distance between each observation and all other observations. (You can use the `**pairwise_distances**` function from **scikit-learn**.) In Vanilla KNN, we calculate the Euclidean distance as shown below:

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

In weighted KNN, the distance is calculated as follows, where for each i-th feature there is a corresponding weight w-i:

$$d(A, B) = \sqrt{\sum_i w_i (A_i - B_i)^2},$$

2. Use the calculations from (1) to select the k nearest neighbors of a particular observation and predict the label of that observation.

3. Based on the predictions, calculate performance metrics like accuracy, AUC, or Log Loss, and return the result. (You are free to choose any of these metrics, but **you are not allowed to use any pre-built functions for calculation**.)

Note that this function takes the weights, dataset, and k as inputs and returns the loss as output.

c) Initialize the weights with zero. Optimize the function from part b to find the best weights that minimize the KNN classification loss. (You can use the `**scipy.optimize.minimize**` function for optimization.)

d) **Implement** the weighted KNN classifier and then use the weights you found in the previous part to predict the labels of the test set. Subsequently, use vanilla KNN to classify the test set and compare the results. [Note that vanilla KNN is the weighted KNN which its weights are all ones.]

e) As you may know, the curse of dimensionality affects the KNN algorithm. Therefore, the fewer the number of features, the better the algorithm performs. We aim to reduce the feature space to only 5 features. Select 8 random subsets of features, each with a size of 5. Then, employ them for classification and report the results.

f) Now, select the 5 features with the highest weights calculated in part c. Then, classify the test set using these features and report the results.

g) Compare random feature selection with no feature selection and weighted feature selection, and justify the results.

Problem 4: Evaluate car safety(25 pts)

#Implementation

In this section, the goal is to create a decision tree classification to predict the safety of vehicles. The dataset includes information on the car's buying price ('buying'), maintenance cost, number of doors ('doors'), seating capacity ('persons'), trunk size ('lug_boot'), and overall safety rating ('safety'). The objective is to build a predictive model that can effectively categorize cars into different safety classes ('class'). By training a Decision Tree Classifier, you'll be able to analyze how these input features influence the safety classification of cars.

a) Load the car_evaluation.csv dataset and check frequency counts of categorical variables.
b) Perform the necessary preprocessing steps.
c) Implement a Decision Tree and report the obtained tree (graph information like depth, leaves, etc.), accuracy, and confusion matric for the test set. (Note: You are only allowed to use "NumPy," "Pandas," and "math" libraries)
d) Implement pruning operation on the decision tree model. Plot a line chart. The Y-axis is the accuracy measure, and X-axis is the max depth of the tree. Discuss obtained chart. (Hint: you can assume a threshold for deciding whether it is valuable to keep a node. The threshold can be applied on one or many metrics like the model`s accuracy in un/pruned states)
e) Repeat part c and d with using sklearn built-in functions and then compare them with your results.