

2021 Introduction to Machine Learning

Program Assignment #2 - Decision Tree & Random Forest & KNN & PCA

TA: Ian Lee ian0908221ian@yahoo.com.tw (mailto:ian0908221ian@yahoo.com.tw)

This programming assignment requires you to understand and implement Decision Tree, Random Forest, K-Nearest Neighbor, and PCA algorithms.

Before we start


Join the discord server for TA support (<https://discord.gg/XJkvmNrcjp>)

- Ask questions on it, and we shall reply.
- Try not to ask for obvious answers or bug fixes.
- Memes and chit-chat welcome.


Objective

1. Data Input - 5%
2. Data Preprocessing - 10%
 - Transform data format and shape so your model can process them.
 - Transform categorical features into **one-hot** representation.
 - **Shuffle the data.**
 - Transform label format so you can do the required two tasks described below Data section.
3. Principal components analysis (PCA) - 25%
 - Compare the results between using the original data and the dimension-reduced one.
 - For dimension reduction, use **Principal components analysis (PCA)**.
 - Reduce the data feature dimension from m to m' where $m' < m$.
 - PCA first calculates the covariance matrix $\text{COV}(x)$.
 - $\text{COV}(x) = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$
 - where $x_i \in \mathbb{R}^{m \times 1}$ is the i -th instance, N is the number of instances, and \bar{x} is the mean of all x_i .
 - Next, it calculates the eigenvectors and eigenvalues of $\text{COV}(x)$.

- $\text{cov}(x) = Q \Lambda Q^T$
- Λ is diagonal matrix of $\text{cov}(x)$ where λ_i is the i-th diagonal element (eigenvalue) in Λ which is ranked by their values in descending order.
- Each column vector of Q is the eigenvector corresponding to a eigenvalue.
- Then choose the top m largest eigenvalues λ_1 to λ_m to be the principal components, and find the corresponding eigenvectors to make Q' .
- Finally, reduce the data dimension by projection using Q'
 - $z_{Ti} = (x_i - \bar{x})^T Q'$
 - where z_i is the projected (dimension reduced) data.

 Note that Q and Q' in PCA must be calculated by the **training set** in order to guarantee the correctness of validation.

4. Model Construction - 15%

 For all the models, you need to do the two tasks described below in the data section.

- The data consists of both **categorical** and **numerical** features, and you have to treat them differently.
- Three models must be constructed, **Decision Tree**, **Random Forest**, and **K-Nearest Neighbor**.
 1. For the Decision Tree model, you may use the following ID3 algorithm pseudocode. - 5%

ID3 (Examples, Target_Attribute, Attributes)

Create a root node for the tree

If all examples are positive, Return the single-node tree Root, with]

If all examples are negative, Return the single-node tree Root, with]

If the number of predicting attributes is empty, then Return the single-node tree Root, with label = most common value of the target attribute in the examples

Otherwise Begin

A ← The Attribute that best classifies examples.

Decision Tree attribute for Root = A.

For each possible value, v_i , of A,

Add a new tree branch below Root, corresponding to the test A

Let Examples(v_i) be the subset of examples that have the value v_i

If Examples(v_i) is empty

Then below this new branch add a leaf node with label = most common value of the target attribute in Examples(v_i)

Else below this new branch add the subtree ID3 (Examples(v_i),

End

Return Root



Note that you could implement any decision tree algorithm, not restricted to ID3. But you need to clarify which algorithm you used. Also, any package-provided model is allowed.

2. For the Random Forest model, you must construct multiple Decision Tree models from randomly selected data (from the training subset) and perform voting for prediction. - 5%
 - For the data selection, the following methods are all acceptable. You could choose one to implement.
 - Randomly select features
 - Randomly select samples
 - Both
 - The number of trees must be greater than or equal to 3. You need to try at least 3 different numbers of trees and compare the result.
 - Understand the difference between K-fold cross-validation and Random Forest. **Confuse one with another, and you won't get this part of the score.**

3. For the KNN model, you need to try at least 3 different K values and compare their results. - 5%

5. Validation - 5%

- Please use 3-fold cross-validation ([https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)#k-fold_cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation)).
- Obtain the final performance by **averaging** all folds' performance.

6. Results - 10%

- Obtain the performances of all experiment settings **in tables** by the following metrics:
 1. Confusion matrix
 2. Accuracy
 3. Sensitivity(Recall)
 4. Precision
- 7. Comparison & Conclusion - 5%
 - Also some feedback, anything you want to tell me.
- 8. Questions - 35% + (10%)
 - Decision Tree
 - Show the **prediction** and **reasoning** of one arbitrary sample in the **testing set**. - 10%
 - Random Forest
 - Describe the difference between **boosting** and **bagging**. - 5%
 - KNN
 - Show the **prediction** and **reasoning** of one arbitrary sample in the **testing set**. - 10%
 - Bonus: pick 2 features, draw and describe the **KNN decision boundaries**. - 10%
 - You can pick 2 features to re-train the model, or just fix every other feature value.
 - PCA
 - In 5-Level classification, reduce the data dimension to 2 using PCA and draw a scatter plot. You have to colorize the data points based on their labels. - 10%

Data - Student Performance Data Set

- Data can be downloaded here:
 - <http://archive.ics.uci.edu/ml/datasets/Student+Performance>
(<http://archive.ics.uci.edu/ml/datasets/Student+Performance>)
- Please **NOTE** that the last column is the label (G3).
- Two datasets provided (Mathematics, Portuguese language) are both acceptable. You could choose one to analyze.
- Followed by this paper (<http://www3.dsi.uminho.pt/pcortez/student.pdf>), You will **have to** do 2 classification tasks:
 - Binary classification - pass if $G3 \geq 10$, else fail.
 - 5-Level classification - based on the Erasmus grad conversion system.

Country	I (excellent/very good)	II (good)	III (satisfactory)	IV (sufficient)	V (fail)
Portugal/France	16-20	14-15	12-13	10-11	0-9
Ireland	A	B	C	D	F

- Data Set Information

- This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features) and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). In [Cortez and Silva, 2008], the two datasets were modeled under binary/five-level classification and regression tasks. Important note: the target attribute G3 has a strong correlation with attributes G2 and G1. This occurs because G3 is the final year grade (issued at the 3rd period), while G1 and G2 correspond to the 1st and 2nd period grades. It is more difficult to predict G3 without G2 and G1, but such prediction is much more useful (see paper source for more details).

- Attribute Information

1. school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
2. sex - student's sex (binary: 'F' - female or 'M' - male)
3. age - student's age (numeric: from 15 to 22)
4. address - student's home address type (binary: 'U' - urban or 'R' - rural)
5. famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
6. Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
7. Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
8. Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
9. Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10. Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11. reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
12. guardian - student's guardian (nominal: 'mother', 'father' or 'other')
13. traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)

14. studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15. failures - number of past class failures (numeric: n if $1 \leq n < 3$, else 4)
16. schoolsup - extra educational support (binary: yes or no)
17. famsup - family educational support (binary: yes or no)
18. paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
19. activities - extra-curricular activities (binary: yes or no)
20. nursery - attended nursery school (binary: yes or no)
21. higher - wants to take higher education (binary: yes or no)
22. internet - Internet access at home (binary: yes or no)
23. romantic - with a romantic relationship (binary: yes or no)
24. famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
25. freetime - free time after school (numeric: from 1 - very low to 5 - very high)
26. goout - going out with friends (numeric: from 1 - very low to 5 - very high)
27. Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
28. Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
29. health - current health status (numeric: from 1 - very bad to 5 - very good)
30. absences - number of school absences (numeric: from 0 to 93)
31. G1 - first period grade (numeric: from 0 to 20)
32. G2 - second period grade (numeric: from 0 to 20)
33. G3 - final grade (numeric: from 0 to 20, output target)

Submission & Scoring Policy

- Please submit a **zip** file, which contains the following, to the newE3 system.
 1. Report
 - Explanation of how your code works.
 - All the content mentioned above.
 - Your name and student ID at the very beginning - 10%
 - Accept formats: **HTML**
 - From markdowns (<https://hackmd.io/?nav=overview>) or jupyter notebooks.
 2. Source codes
 - Accept languages: **python3**
 - Accept formats: **.ipynb** (<https://jupyter.org/>)
 - Package-provided models are allowed
- Your score will be determined mainly by the submitted report.

- If there's any problem with your code, TA might ask you (through email) to demo it. Otherwise, no demo is needed.
- Scores will be adjusted at the end of the semester for them to fit the school regulations.
- **Plagiarizing is not allowed.**
 - Plagiarizing is checked by MOSS (<https://theory.stanford.edu/~aiken/moss/>) and manually afterward.
 - You will get **ZERO** on that homework if you get caught the first time.
 - The second time, you'll **FAIL** this class.
 - 抄襲第一次作業零分，第二次當掉

Acknowledgments

- Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml> (<http://archive.ics.uci.edu/ml>)]. Irvine, CA: University of California, School of Information and Computer Science.

Tools that might be useful

- Jupyter Lab (<https://jupyter.org/>) - Better data science experience
- numpy (<https://numpy.org/>) - Math thingy
- matplotlib (<https://matplotlib.org/tutorials/introductory/pyplot.html>) - Plot thingy
- pandas (<https://pandas.pydata.org/>) - Data thingy
- scipy (<https://www.scipy.org/>) - Science thingy
- scikit-learn (<https://scikit-learn.org/stable/>) - Machine Learning and stuff