- Due Jun 28, 2023 by 11:59pm
- Points 100
- Submitting a file upload
- Available until Jul 1, 2023 at 11:59pm

This assignment was locked Jul 1, 2023 at 11:59pm.

A5 Blackbox methods, KNN

Instructions

Input files: ***NA_sales_filtered.csv*** *(https://utah.instructure.com/courses/882385/files/147060997/download?wrap=1)* *(https://utah.instructure.com/courses/882385/files/147060997/download?download_frd=1)* video game sales dataset from Assignment 4. The target variable is still NA_Sales for numeric prediction using Blackbox methods.

Packages required: Install caret, RWeka, kernlab, rminer, matrixStats, and knitr packages.

Submission:

Submit two files – A5_yourLastName_yourFirstName.Rmd and A5_yourLastName_yourFirstName.html, (or another output format) generated from rendering (or knitting) A5_yourLastName_yourFirstName.Rmd.

(Please do not compress these two files into one zip file. Submit them as separate files. Be sure that the HTML output file contains assignment title, author name – you, the file creation date, and table of content including section numbers.)

Note: If there is any discrepancy in the task descriptions in the assignment page and the short rubric descriptions, please follow the task descriptions in the assignment page.

Task 1

Create A5_yourLastName_yourFirstName.Rmd to meet the following requirements:

Code chunk 1- Package load, data import, inspection, and partitioning (10%)

A. Load all the required packages

B. Import the NA_sales_filtered.csv and partition the dataset to the training set and testing set

i. Import NA_sales_filtered.csv and set stringsAsFactors = False.

ii. Create a data frame with all of the variables except for *Name.*

iii. Transform character variables except for Name to factors.

iv. Create the training and testing sets based on percentage split – 70% for training and 30% for testing.

Code chunk 2 - Build and evaluate neural network models for numeric prediction tasks (20%)

A. Build and evaluate MLP models for numeric prediction with the video game sales data (imported and prepared in 1B).

i. Build an MLP model on MultilayerPerceptron()'s default setting on the training set. Evaluate the model performance on the training set and testing set.

ii. Build a two-hidden-layer MLP model and change one of the other hyper-parameter values – e.g. the learning rate on the training set. Evaluate the model performance on the training set and testing set.

Code chunk 3 - Build and evaluate SVM (ksvm) models for numeric prediction tasks (20%)

A. Build and evaluate ksvm models for numeric prediction with the video game sales data (imported and prepared in 1B).

i. Build a model on ksvm()'s default setting on the training set. Evaluate the model performance on the training set and testing set.

ii. Build a ksvm model using a different kernel function on the training set. Use the default C value. Evaluate the model performance on the training set and testing set.

iii. Build a ksvm model using a different cost value (i.e. C= c, where c>1) on the training set. Evaluate the model performance on the training set and testing set.

Code chunk 4 - Build and evaluate knn (IBk) models for numeric prediction tasks (20%)

A. Build and evaluate IBk models for numeric prediction with the video game sales data (imported and prepared in 1B).

i. Build a model on IBk()'s default setting on the training set. Evaluate the model performance on the training set and testing set.

ii. Build an IBk model using a different K value on the training set. Hold other parameters at the default setting. Evaluate the model performance on the training set and testing set.

iii. Build an IBk model using a weighted voting approach (e.g. I=TRUE) on the training set. Evaluate the model performance on the training set and testing set.

iv. Build an IBk model by automatically selecting K (i.e., X=TRUE) on the training set. Evaluate the model performance on the training set and testing set.

Code chunk 5 - Cross-validation function for numeric prediction models (10%)

A. Define a named function (e.g., cv_function) for cross-validation evaluation of classification or numeric prediction models with df, target, nFolds, seedVal, method and metrics_list for input.

B. Generate a table of fold-by-fold performance metrics, and means and standard deviations of performance over all of the folds.

Code chunk 6 - 3 fold cross-validation of MLP, ksvm and IBk models (15%)

A. Use the default settings of MultilayerPerceptron(), ksvm and IBk to perform cross-validation for numeric prediction with the video game sales data (imported and prepared in 1B).

Others:

- Add some simple descriptive text in the text area before the code chunk.
- Add a name or description of each code chunk in {r}. Make sure that you include all the code and output from executing code when rendering A5_yourLastName_yourFirstName.Rmd to HTML.
- Feel free to add comment lines with the requirement item numbers (e.g., # 3.A or # 3.B) to your code cell to help TAs and instructors easily identify your code that addresses a particular requirement.

Task II Reflections(5%):

What have you learned from building each of these models and the modeling impact of your adjustments to the hyperparameters or dataset? If you were explaining the results of these models to a supervisor what would you say about them? Attempt to do more than just state facts here, interpret the results. Coding is great, interpretation of output is even more important. Discuss each model. Write at least 150 words.

Use the RMD header to include assignment title, author name – you, and the file creation date. Also, include header specifications to generate a table of contents and section numbers of your code chunks.

Render A5_yourLastName_yourFirstName.Rmd to HTML output.

Submit two files – A5_yourLastName_yourFirstName.Rmd and A5_yourLastName_yourFirstName.html, (or another output format) generated from rendering (or knitting) A5_yourLastName_yourFirstName.Rmd.

(Please do not compress these two files into one zip file. Submit them as separate files. Be sure that the HTML output file contains assignment title, author name – you, the file creation date, and table of content including section numbers.)

Assignment-4 Rubric

Criteria	Ratings			Pts
1.A. Load all required packages in RStudio. Use getwd() and setwd() to the current working directory in RStudio.	**5 pts** **Full Marks**	**5 to >0.0 pts** **Partially**	**0 pts** **No Marks**	5 pts
1.B.i. Import NA_sales_filtered.csv and set stringsAsFactors = False.	**1 pts** **Full Marks**	**1 to >0.0 pts** **Partially**	**0 pts** **No Marks**	1 pts
1.B.ii. Create a data frame with all of the variables except for Name.	**1 pts** **Full Marks**	**1 to >0.0 pts** **Partially**	**0 pts** **No Marks**	1 pts
1.B.iii. Transform character variables to factors.	**1 pts** **Full Marks**	**1 to >0.0 pts** **Partially**	**0 pts** **No Marks**	1 pts
1.B.iv. Create the training and testing sets based on the target variable (70% for training and 30% for testing).	**2 pts** **Full Marks**	**1 pts** **Partially**	**0 pts** **No Marks**	2 pts
2.A.i. Build a MLP model by default setting on the training set for NA_sales_filtered data. Evaluate the model performance on the training set and testing set.	**10 to >7.5 pts** **Full Marks**	**7.5 to >0.0 pts** **Partially**	**0 pts** **No Marks**	10 pts
2.A.ii. Build a two-hidden-layer MLP model and change one of the other parameter values. Evaluate the model performance on the training set and testing set.	**10 to >8.33 pts** **Full Marks**	**8.33 to >0.0 pts** **Partially**	**0 pts** **No Marks**	10 pts
3.A.i. Build a ksvm model by default setting on the training set for NA_sales_filtered data. Evaluate the model performance on the training set and testing set.	**6 to >4.0 pts** **Full Marks**	**4 to >0.0 pts** **Partially**	**0 pts** **No Marks**	6 pts
3.A.ii. Build a ksvm model using a different kernel function on the training set. Use the default C value. Evaluate the model performance on the training set and testing set.	**8 to >6.0 pts** **Full Marks**	**6 to >0.0 pts** **Partially**	**0 pts** **No Marks**	8 pts
3.A.iii. Build a ksvm model using a different cost value on the training set. Evaluate the model performance on the training set and testing set.	**6 to >4.0 pts** **Full Marks**	**4 to >0.0 pts** **Partially**	**0 pts** **No Marks**	6 pts
4.A.i. Build an IBK model by default setting on the training set for NA_sales_filtered data. Evaluate the model performance on the training set and testing set.	**5 to >4.0 pts** **Full Marks**	**4 to >0.0 pts** **Partially**	**0 pts** **No Marks**	5 pts
4.A.ii. Build an IBk model using a different K value on the training set. Hold other parameters at the default setting. Evaluate the model performance on the training set and testing set.	**5 to >4.0 pts** **Full Marks**	**4 to >0.0 pts** **Partially**	**0 pts** **No Marks**	5 pts
4.A.iii. Build an IBk model using a weighted voting approach (e.g. I=TRUE) on the training set. Evaluate the model performance on the training set and testing set.	**5 to >4.0 pts** **Full Marks**	**4 to >0.0 pts** **Partially**	**0 pts** **No Marks**	5 pts
4.A.iv. Build an IBk model by automatically selecting K (i.e., X=TRUE) on the training set. Evaluate the model performance on the training set and testing set.	**5 to >4.0 pts** **Full Marks**	**4 to >0.0 pts** **Partially**	**0 pts** **No Marks**	5 pts
5.A. Define a named function (e.g., cv_function) for cross-validation evaluation of classification or numeric prediction models with df, target, nFolds, seedVal, method and metrics_list for input.	**7 to >6.0 pts** **Full Marks**	**6 to >0.0 pts** **Partially**	**0 pts** **No Marks**	7 pts

Criteria	Ratings			Pts
5.B. Generate a table of fold-by-fold performance metrics, and means and standard deviations of performance over all folds.	**3 to >2.0 pts** **Full Marks**	**2 to >0.0 pts** **Partially**	**0 pts** **No Marks**	3 pts
6.A. Use the default settings of MultilayerPerceptron(), ksvm and IBk to perform cross-validation for numeric prediction by NA_sales_filtered data.	**15 to >14.0 pts** **Full Marks**	**14 to >0.0 pts** **Partially**	**0 pts** **No Marks**	15 pts
What have you learned from building each of these models and the modeling impact of your adjustments to the hyperparameters or dataset? If you were explaining the results of these models to a supervisor what would you say about them? Attempt to do more than just state facts here, interpret the results. Coding is great, interpretation of output is even more important. Discuss each model. Write at least 150 words.	**5 to >4.0 pts** **Full Marks**	**4 to >0.0 pts** **Partially**	**0 pts** **No Marks**	5 pts
				Total Points: 100