

Introduce file

Code file:

Filename: seg.py

Usage: Drop the columns which the test.csv file does not provide('Step Start Time', 'First Transaction Time', 'Correct Transaction Time', 'Step End Time', 'Step Duration (sec)', 'Correct Step Duration (sec)', 'Error Step Duration (sec)', 'Incorrects', 'Hints', 'Corrects'). And separate the file into different part with set the number of lines.

Command: python3 seg.py

Filename: asc_encode.py

Usage: Using ascii encoding to transform string data into integer data. For every single letter in a string, transform it into ascii number and join them all together.

Command: python3 asc_encode

Filename: ordinary_encode.py

Usage: Using bag encoding to transform string data into integer data. (Decision tree care less about the continuity, so bag encoding is enough)

Command: python3 ordinary_encode.py filename (without .csv)

Filename: clf.py

Usage: Model implementation with decision tree (classification). The model accepts 2 files: train file and test file. Files will be automatically divided into data and target. In the bottom of clf.py, codes used to test the hyperparameters are commented. Clf.py will generate the model and save it to clf.m file for further use.

Command: python3 clf.py

Filename: rfc.py

Usage: Model implementation with random forest. The model accepts 2 files: train file and test file. Files will be automatically divided into data and target. In the bottom of rfc.py, codes used to test the hyperparameters are commented. Rfc.py will generate the model and save it to rfc.m file for further use.

Command: python3 rfc.py

Filename rgr.py

Usage: Model implementation with decision tree (linear regression). The model accepts 2 files: train file and test file. Files will be automatically divided into data and target. In the bottom of rgr.py, codes used to test the hyperparameters are commented. Rgr.py will generate the model and save it to rgr.m file for further use.

Command: python3 rgr.py

Filename: svm.py

Usage: Model implementation with Support Vector Machine (SVM). The model accepts 2 files: train file and test file. Files will be automatically divided into data and target. In the bottom of svm.py, codes used to test the hyperparameters are commented. Rgr.py will generate the model and save it to svm.m file for further use.

Command: python3 svm.py

Filename: model_figure.py

Usage: Show the given decision tree with figure. And display the importance of every feature in the model

Command: python3 model_figure.py

Data File

Filename: smalltest.csv

Description: 200 rows data from train.csv

Filename: smalltest_oh.csv

Description: 200 rows data from train.csv. Encoding with ordinary encoding.

Filename: smalltest_trans.csv

Description: 200 rows data from train.csv. Encoding with ascii encoding.

Filename: smalltrain.csv

Description: 2000 rows data from train.csv

Filename: smalltrain_oh.csv

Description: 2000 rows data from train.csv. Encoding with ordinary encoding.

Filename: smalltrain_trans.csv

Description: 2000 rows data from train.csv. Encoding with ascii encoding.

Filename: wholetest.csv

Description: 2000 rows data from train.csv

Filename: wholetest_oh.csv

Description: 2000 rows data from train.csv. Encoding with ordinary encoding.

Filename: wholetest_trans.csv

Description: 2000 rows data from train.csv. Encoding with ascii encoding.

Filename: wholetrain.csv

Description: 200000 rows data from train.csv

Filename: wholetrain_oh.csv

Description: 200000 rows data from train.csv. Encoding with ordinary encoding.

Filename: wholetrain_trans.csv

Description: 200000 rows data from train.csv. Encoding with ascii encoding.

Filename: Finaltest.csv

Description: 600 rows data from train.csv

Filename: Finaltest_oh.csv

Description: 600 rows data from train.csv. Encoding with ordinary encoding.

Filename: Finaltest_trans.csv

Description: 600 rows data from train.csv. Encoding with ascii encoding.

Filename: Answerofproject.zip

Description: Fill in Nan of "Correct First Attempt". The answer of the project.