System Requirements Specification Index

For

Detect Whether A Car Is A Good Buy Or Not

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Problem Statement : Provide a code solution to predict whether an auction is a kick or not.

Description : One of the biggest challenges of an auto dealership purchasing a used car at an auto auction is the risk that the vehicle might have serious issues that prevent it from being sold to customers. The auto community calls these unfortunate purchases "kicks". The challenge of this competition is to predict if the car purchased at the Auction is a Kick (bad buy).

Data is given to you in the code. You need to implement all the methods given in ml.py. Please adhere to the instructions below and try not to change any method API. Doing so will result in a failed submission.

The solution contains the following folder structure.

dont_get_kicked |

|--code

|--__init__.py

|--ml.py – your code goes here

|--constants.py – defines few constants

|--data

|--data.csv – data file for the problem

|--data_description.txt – description of the data labels/columns

|--model – folder where you save your trained model

|--graphs – folder where you save graphs of your analysis

|--tests – contains unit testcases for the solution

|--docs – contains documents

|--dont_get_kicked.xml – for testing

|--Don't Get Kicked Document.docx – this document

|-- requirements.txt

The ml.py has the following class.

class model():

def __init__():

--self.data_file = absolute path of the data file, please don't use any external source.

--self.model_file = absolute path of the final model file. Please see we will only consider the model saved with a given model_file name for evaluation. You are free to experiment with multiple/any model(s).

--self.graphs_folder = absolute path of folder where you save your graphs of your analysis. You can save multiple files.

def data_transformation(self, test_data=None, is_train=True):

--Is basically the method when given raw input data returns the final transformed data. Please see, It can be used in two ways:

1. while training:

called as data_transformation(test_data=None, is_train=True) that is no test data is passed and you access train data from self.train file and perform the transformations on the data and return either return X_train, X_test, y_train, y_test or return X_train, X_valid, X_test, y_train, y_valid, y_test

2. while testing:

called as data_transformation(test_data=test_data, is_train=False) and you perform the same transformations performed on the train data i.e., self.train but return only transformed test data ie., return test_data_transformed.

def model_fit(self, X_train, y_train):

-- Takes X_train, and y_train and fit a model of your choice. Do not forget to save the final model at location self.model_file.

def model_predict(self, X_test):

-- Takes X_test and returns the predicted value of the saved model at self.model_file.

def cost_metric(self, y_true, y_pred):

-- Given y_true and y_pred return the value of your choice of cost metric.

Execution Steps to Follow:

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
3. This editor Auto Saves the code
4. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
6. To setup environment:

   pip install -r requirements.txt --user

   pip install requests

   pip install pandas

   pip install numpy

   pip install sklearn

7. To launch application:

   python3 code/ml.py

8. To run Test cases:

   python3 -m pytest tests

9. Before Final Submission also, you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository for code quality analysis graph.

-----x-----