EECS 461/ECE 523
MACHINE LEARNING
Fall 2019

ASSIGNMENT 1

Due Date: *Tuesday, November 19th, 2019, 23:59*

Assignment Submission:

1. Turn in your assignment by the due date through LMS**.**
2. Prepare a single Jupyter Notebook (.ipynb) with the answers to all questions. **Name the file as <your first name>_<your last name>_ assignment1.ipynb.**
3. Make sure to **use the sample Jupyter Notebook file provided to you as template**.

All work in questions must be your own; you must neither copy nor provide assistance to anybody else. If you need guidance for any question, talk to the instructor or TAs in office hours. You can also reach your TAs via email:

- Zeina Termanini at zenatermanini@std.sehir.edu.tr
- Mohammad Abunada at mohammedabunada@std.sehir.edu.tr

Late Assignment Policy: You have a total of **4 days of late assignment** turn-in allowance throughout this semester. For a single assignment, you can use **a maximum of 2 late-days.** You decide which assignments you are going to use your 4 late-days. After assignment due date/time, each 24-hours period is counted as one late date (i.e., if you submit your assignment 1 hour late or 23 hours late, you use 1 late-date). It is your responsibility to keep track of your late days. If you are late more than 2 days for any assignment or you exhausted your late days, you get 0 from the late assignment **(No exceptions)**

Assignment Overview:

In this assignment, you will be running exploratory analysis on a dataset to better understand it and its features. You will be processing and preparing the data to apply the machine learning knowledge you've obtained through the lectures. This will include creating, analysing and generating predictions with regression models. This assignment is mainly about the examples in the chapter 2, 3 and 4 of the course book with a different data set. Reviewing the book and the corresponding code will greatly help you. **You are expected to primarily use Scikit-Learn in the assignment.**

Data Set:

The data set provided for this assignment contains information on many different car types and their prices. This assignment challenges you to ***predict the sale price of each car***. Data is in CSV format and has already been split into training and test sets for your convenience: train.csv: the training set, test.csv: the test set.

DATA PREPARATION (25 points)

In the first part of the assignment, you will analyze the dataset and preprocess it in order to prepare it for using machine learning algorithms. In this data set, our target variable is "price" while the others are our features.

(a) (5 points) Split your data into X and y:

As mentioned, "price" column is our dataset target. Create two pandas data frames using train.csv, one containing all the input features and the other containing the target label only. Name these data frames as **train_x_a** and **train_y** respectively.

(b) (5 points) Handling missing values:

Find all features (columns) that contain missing (NaN) values. Store these column names in a list called **nan_columns**. Fill the missing values with the median value of the corresponding feature. Save your resulting data frame as **train_x_b.** (Note that if there are any missing values in the target i.e. price column, drop the corresponding row completely from train_x and train_y)

(c) (5 points) Handling categorical variables:

Find all features (columns) that contain categorical values (strings). Store these column names in a list called categorical_columns. (For example, if the data has a column titled ‘gender’ with ‘female’ and ‘male’ values, then ‘gender’ should be in the **categorical_columns** list.)

(d) (5 points) One hot encoding:

Perform ***one hot encoding*** on features with categorical values. Modify train_x_b by replacing categorical columns with their one-hot encoding representations. Name your modified dataframe as **train_x_d .** (For example, if you have a column named “gender” that has two unique values (male and female), after one-hot encoding, gender column will be replaced with two new columns in the dataframe, one column for male and one column for female. Your new dataframe will have 1 in the female column and 0 in the male column.)

(e) (5 points) Standard scaling:

Scale all columns in train_x_d with standardization. Name the new dataframe with scaled values as **train_x_e**. (Note: Sk Learn provides a transformer called StandardScaler for standardization. The output of the scaler is a numpy array. You need to convert it dataframe after standardization. Don’t forget to add the original data frame’s indices and columns to the new data frame.)

DATA EXPLORATION (15 points)

In this part of the assignment, you are going to calculate and visualize certain features of your dataset to understand it better. This is an important step before modelling to find out any problems there might be with your data.

(f) (5 points) Visualize variable distributions:

Plot the histogram of each of the variables in your dataset. Try to understand how each variable is distributed. Are there any extreme points in these distributions?

(g) (5 points) Calculate correlation:

Calculate the correlation score between all continuous variables in your dataset and the target. Get the strongest 5 correlating variables (top 5 **absolute** correlations) and store the names in a list called **top_5_corr.**

Note: *Correlation will help you understand the relation between a variable and the target. If they move in the same direction (up or down) at the same time, they have high correlation.*

(h) (5 points) Scatter plot:

Plot each of the 5 variables you found in the previous question **(f)** against each other. These plots should form a matrix with 5 x 5 plots.

LINEAR REGRESSION TO PREDICT CAR PRICES (60 points)

In this part of the assignment, you are going to train a Linear Regression model that predicts the prices of cars by using the other features in the dataset. When asked to retrieve the MSE score from cross_val_score or GridSearchCV, set the scoring option to 'neg_mean_square_error'. This will return a negative error. **To obtain MSE, get the absolute value.**

(i) (5 points) Create a model:

Create a Linear Regression model with default parameters. Train the model with train_x_e and train_y. Print the Mean Square Error (MSE) for training data.

(j) (5 points) Validate your model:

Perform 5-fold cross validation with training data and print MSE score for each fold and their average (mean)

(k) (5 points) Test your model:

Using test.csv, create **test_x** that has all the features except our target "price" and **test_y** that has only "price". Fill missing values in test_x with median value, perform feature scaling and apply one hot encoding to categorical values, same as what you did in the first part of the assignment. (Note that transformations you apply to the test set **should be same** as the one applied to training set. When filling the missing values, the median value should be the one computed for the training set. Similarly, in standardization, the mean and the standard deviation should be the ones from the training set.)

(l) (5 points) Predict on test set:

Predict the prices of cars in ***test_x*** data using your linear regression model that you created in **(i)**. Store the predicted values in a variable named **predicted_values**. Print the test set MSE of your model. Also, print you model's coefficients.

(m) (10 points) Polynomial Features:

Some of the features within the dataset may have a polynomial relation with the target. In order to account for this, run a polynomial transformation with degree 2 on **train_x_e** and store the result in a variable called **train_x_m.** Create a Linear Regression model with default parameters and perform 5-fold cross validation using **train_x_m** and **train_y**. Print the average MSE score.

(n) (10 points) Regularization:

Now that many more features are incorporated into the training set, there is a high chance our new model is overfitting the training set. Create a Lasso regularization model with default parameters. Perform 5-fold cross validation using the training data and print the average MSE score.

(o) (10 points) Regularization Curve:

Calculate and store all 5-fold CV MSE scores for alpha values between 1 and 3000 with a step size of 10. Plot the scores as a line graph and print the lowest error and it's respective alpha.

(p) (10 points) Grid Search:

You decided Linear regression was not good enough and decided to use Support Vector Machines (SVMs). SVMs have many hyper parameters which you may not be familiar with. So you decide to use grid search to find the best set of parameters to use. Using **train_x_e** and **train_y**, run 5-fold grid search CV (set cv=5) using sklearn's grid search function on sklearn.svm.SVR with the following search parameters:

- Kernel: linear, C: 10.0, 30.0, 100.0, 300.0, 1000.0, 3000.0, 10000.0, 30000.0
- Kernel: rbf (radial), C: 1.0, 3.0, 10.0, 30.0, 100.0, 300.0, 1000.0, 3000.0 and gamma: 0.01, 0.03, 0.1, 0.3, 1.0, 3.0

Print out the best MSE score and best hyperparameters found.

IMPORTANT NOTES

- Prepare and upload one Jupyter notebook file, which should be named as <your first name>_<your last name>_ assignment1.ipynb.
- A template Jupyter notebook file provided to you. Follow the template's structure.
- Explain your code with comments.
- **Plagiarism in any form will not be tolerated. Changing variable names is not solving an assignment.**

Wrong file name format	-10 points
Not using template	-10 points
Not using correct variable (dataframe) names	-10 points
Insufficient comments	**Any part with insufficient comments will not be graded.**