EECS 461/ECE 523
MACHINE LEARNING
Fall 2019

ASSIGNMENT 3

Due Date: *Friday, December 27th, 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>_ assignment3.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)**

*** Please include a thorough description of your problem. If you are having problems with your code, include a screenshot of the exact problem. Please don't enquire if your answer is correct or not.**

Assignment Overview:

This assignment is mainly about the examples in chapters 6 and 7 of the course book with a different data set. Reviewing the book and the corresponding code will help you. **You are expected to primarily use Scikit-Learn in the assignment.**

Data Set:

The data set provided for this assignment is the same as the one assigned in the first one. It 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

Follow the same steps from the first assignment to create the variables **train_x_e, train_y, test_x, and test_y.**

KNN REGRESSOR TO PREDICT CAR PRICES (25 points)

In this part of the assignment, you are going to train a KNN Regression model that predicts the prices of cars by using the other features in the dataset.

(a) (15 points) Training vs Validation Plot:

Split your training dataset **(train_x_e and train_y)** into a validation and new training set (**80% training and 20% validation**, use **random_state = 0** with sklearn's train_test_split). For every integer value **k between 1 and 30** create and record a KNN Regression model's **training** and **validation** MSEs where the KNN model's number of neighbours is k. Produce a plot that visualizes both your training and validation errors over the differing values of k: where the y-axis is the error and the x-axis is k.

(b) (10 points) Test your model:

For the best k you found in (a), re-create your best model as best_knn and report it's test MSE on the test set.

DECISION TREE REGRESSOR TO PREDICT CAR PRICES (25 points)

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

(c) (15 points) Grid Search to find best model:

As decision trees have large hyper parameter sets, you will be using gridsearch to find the best combinations. Using the below parameters, run GridSearchCV (cv = 5) with a Decision Tree model on **train_x_e** and **train_y.** Print out the best model’s parameters and error. **Make sure you set random_state to 0.**

Parameter Name	Parameter Values
max_depth	None, 1, 5, 10
min_samples_split	0.01, 0.05, 0.1, 0.3
max_features	auto, sqrt, log2
max_leaf_nodes	10, 50, 100, 250
random_state	0

(d) (10 points) Test your model:

Recreate your best Decision Tree model from (c) as best_tree and report it’s test MSE on the test set.

ENSEMBLE BAGGING REGRESSOR TO PREDICT CAR PRICES (25 points)

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

(e) (15 points) Grid Search to find best model:

In order to utilize bagging ensembling in sklearn you will be working with sklearn.ensemble.BaggingRegressor. Using the below parameters, run GridSearchCV (cv = 5) on **train_x_e** and **train_y.** Print out the best model’s parameters and error.

Make sure you set random_state to 0.

Parameter Name	Parameter Values
base_estimator	LinearRegression(), KNeighboursRegressor(), DecisionTreeRegressor(random_state=0)
n_estimators	25, 50, 100, 250
bootstrap_features	False, True
random_state	0

(f) (10 points) Test your model:

Recreate your best Ensemble Bagging model from (e) as best_bag and report it’s test MSE on the test set.

RANDOM FOREST REGRESSOR TO PREDICT CAR PRICES (25 points)

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

(g) (10 points) Grid Search to find best model:

In order to utilize random forests in sklearn you will be working with sklearn.ensemble.RandomForestRegressor. Using the below parameters, run GridSearchCV (cv = 5) with a Random Forest model on **train_x_e** and **train_y.** Print out the best model's parameters and error. **Make sure you set random_state to 0.**

Parameter Name	Parameter Values
n_estimators	25, 50, 100, 250
max_depth	None, 1, 5, 10
random_state	0

(h) (10 points) Test your model:

Recreate your best Random Forest model from (g) as best_rf and report it's test MSE on the test set.

(i) (5 points) Feature Importances:

Print out the three most important features according to your best model.

IMPORTANT NOTES

- Prepare and upload one Jupyter notebook file, which should be named as <your first name>_<your last name>_ assignment3.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.**