Exercise 3

This exercise is not for submission.

However it is highly recommended that you do it as preparation for the exam

In this exercise you will practice object oriented & functional programming while practicing data science and machine learning tasks.

About the data

In the file 'data.csv', you will find information about residential homes deals made in Ames, Iowa.

Each deal is summarized in a row, which includes the property sale price along with 79 characteristic of the property (and its region).

Just like in any typical data-science project, you will need to:

- Perform data pre-processing tasks
- Perform feature selection
- Apply machine learning algorithms

The file data_description.txt includes a full description of the data.

About the module design

The code in the skeleton file is organized in 7 classes:

1. `DataManager`: reads the data, and provides methods for handling the data (i.e., data selection, filtration, transformation and more)
2. `CorrelationManager`: provides utility functions for feature selection based on Pearson's correlation
3. `Learner` (abstract class) - common learning algorithm API: supports model training, computes model score and outputs model summary
4. `RegressionLearner` (abstract class) - common regression algorithm API: inherits from `Learner` and includes methods specific to regression based model
5. `LinearRegressionLearner`: inherits from `RegressionLearner` and uses a linear regression model
6. `LogisticRegressionLearner` (abstract class): inherits from `RegressionLearner` and uses a logistic regression model
7. `DT`: inherits from `Learner` and uses a decision tree model

To architecture above is designed to prevent code duplication and provide a common API to the different learners. Try to avoid as much code duplication as possibly by implementing common behaviors in the parent's class.

Notes about the skeleton:

- The skeleton file includes a main function, use it and fill the code under the function declarations
- Do not change the main function as it orchestrates all tasks

- pd: refers to the pandas module
- np: refers to the numpy module
- df: refers to an instance of pandas.core.frame.DataFrame
- sr: refers to an instance of pandas.core.series.Series
- sns: refers to import seaborn as sns
- plt: refers to import matplotlib.pyplot as plt
- To simplify things, in this exercise, you are not required to handle errors, and may assume that the input parameters and usage ordering are correct
- In all functions, make sure to return the correct types (e.g., do not confuse a list and pd.Series)

The main function uses the classes described above to read the data, investigate it, select predictive features, and run several ML algorithms.

First it uses the DataManager to read and investigate the characteristic (features) of houses.

Then, it uses the CorrelationManager to select the top features to use during the learning phase.

Lastly, it uses linear regression, logistic regression and decision tree to explore the relationship between the numerical house characteristics (independent variables) and the following target attributes:

- 'SalePrice' attribute
- 'LuxuryApparent' attribute

Note, that 'LuxuryApparent' attribute does not appear in the original data (i.e, it is a derived attribute). We consider any house that is more expensive than 90% of the houses as a 'luxury apparent'.

For the main to be fully functional, you are requested to complete the missing code in all the #TODO statements.

At the top of each function, you will find a complete description of the input/output parameters and an implementation guidance (you may follow it, but can also implement things 'your way')

Once you are done, use the official solution to check the correctness of your results.

Importantly, when you are done ensure that you can answer the following questions:

1. How does the distribution of the 'SalePrice' look like (i.e., range, does it resemble any well-known distribution, is the data concentrated around one or several centres)?
2. Why did we remove features that had high correlation with each other?
3. What is the meaning of the logistics/linear regression model scores? coefficients? p-values?
4. What is the interpretation of the decision tree model? its score? its cross validation score?
5. In the regressions, were the smaller or bigger models better (use the model summary and scores to decide)?

Good luck!