1**Select the correct option for the following statements**

All other answers are incorrect

5% of the dataset present null values

10% of the dataset present null values

1% of the dataset present null values

SubmittedCorrect!

Start project before checking your activities.

2Are SALE\_PRC and Age the same kind of data?

Yes

No

SubmittedCorrect!

Start project before checking your activities.

3**Select the correct option for the following statements**

There are 9 continuous variables.

Structure\_quality is a discrete variable.

Age is a continuous variables in the dataset.

SubmittedCorrect!

Start project before checking your activities.

4 **Is the dataset likely to have duplicated values?**

No

Yes

SubmittedCorrect!

Start project before checking your activities.

Calculate the correlation matrix and answer the following questions.

5**Which variable presents the highest correlation with 'SALE\_PRC'?**

LONGITUDE

RAIL\_DIST

OCEAN\_DIST

TOT\_LVG\_AREA

SubmittedCorrect!

Start project before checking your activities.

6

**What is the correlation between 'structure\_quality' and 'SALE\_PRC'?**

*Round to two decimal places*



SubmittedCorrect!

Start project before checking your activities.

7.**Is the 'SALE\_PRC' positive associated to the 'OCEAN\_DIST'?**

Yes

No

SubmittedCorrect!

Start project before checking your activities.

8 **What is the correct figure to represent the quantity of these variables?**

*There could be more than just one correct answer.*

sns.barplot(x="features", data=df)

sns.countplot(x="features", data=df)

sns.catplot(x="features", data=df, kind="count")

sns.catplot(x="features", data=df)

SubmittedCorrect!

Start project before checking your activities.

9 **Scatterplot**

Plot the relation between SALE\_PRC and structure\_quality, SPEC\_FEAT\_VAL and TOT\_LVG\_AREA using the following code.

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 3,sharey=True, figsize=(16,8))

sns.scatterplot(x = df['structure\_quality'],y = df['SALE\_PRC'], ax=ax1)

sns.scatterplot(x = df['SALE\_PRC'],y = df['SALE\_PRC'], color='tab:orange', ax=ax2)

sns.scatterplot(x = df['TOT\_LVG\_AREA'],y = df['SALE\_PRC'], color='tab:green', ax=ax3)

sns.scatterplot(x = df['OCEAN\_DIST'],y = df['SALE\_PRC'], color='tab:red', ax=ax4)

plt.show()

Copy

Can you find the mistake in this code?

The number of cols should be 2

The figure in ax2 should be replace for : sns.scatterplot(x = df['structure\_quality'],y = df['SALE\_PRC'], color='tab:orange', ax=ax1)

The figure in ax2 should be replace for : sns.scatterplot(x = df['SPEC\_FEAT\_VAL'],y = df['SALE\_PRC'], color='tab:orange', ax=ax2)

SubmittedCorrect!

Start project before checking your activities.

10.**Histogram**

Plot the histogram of the numerical variables in one figure using bin size equal to 20.

Write the missing word of the code below:

df.---(bins = 20, figsize = (20,20), color = 'b');

Copy



SubmittedCorrect!

Start project before checking your activities.

11.**Outliers**

Some observations within a set of data may fall outside the general scope of others ones. Such observations are called **outliers**. Which plot could you use to identify them?

Boxplot

Histogram

Barplot

Scatterplot

SubmittedCorrect!

Start project before checking your activities.

12**Drop features**

Remove the following features LATITUDE, LONGITUDE, and PARCELNO. For this task select the correct answer

df.dropna(columns=['LATITUDE', 'LONGITUDE','PARCELNO'],inplace=True)

df.drop(columns=['LATITUDE', 'LONGITUDE','PARCELNO'],inplace=True)

df.drop(columns=['LATITUDE', 'LONGITUDE','PARCELNO'])

df= df.drop(columns=['LATITUDE', 'LONGITUDE','PARCELNO'])

SubmittedCorrect!

Start project before checking your activities.

13**Outliers**

*As we can see the dataset has a large number of outliers in them.*

Outliers can be handled using some outlier handling techniques or else we can use Machine learning models that are robust to outliers.

Complete a function using the **IQR** method to identify outliers and return the percentage per variable:

**def** **IQR\_outliers**(data,limit=---):

numColumns = ----.select\_dtypes(include=-----).columns.tolist(); # extract list of numeric columns

Q1 = -------

Q3 = ------

IQR = ------- - ------;

outliers=((data[numColumns] < (-- - limit\*---)) | (data[numColumns] > (-- - limit\*---))).**sum**()\*---/data.shape[-]

**return** outliers

SALE\_PRC has 9%

WATER\_DIST has 20%

SPEC\_FEAT\_VAL has 12%

LONGITUDE has 1%

SubmittedCorrect!

Start project before checking your activities.

14

**Classification or Regression**

The objective of this work is to construct a satisfying predicting model to forecast Miami housing prices that provide reliable references for people in Miami. Based on this, you should select wheater this scenario is a classification or a regression problem.

Regression

All other answers are incorrect

Classification

SubmittedCorrect!

Start project before checking your activities.

15.**Target and Feature Separation**

Separate the target and the features in two variables. Complete the missing word of the code in the following box.

X=df.drop(['------'],axis=1)

Y=df['-----']

Copy



SubmittedCorrect!

Start project before checking your activities.

16.**Train and Test**

Select the correct way to split the dataset in 20% test and 80% train.

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.80, random\_state=0)

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.70, random\_state=0)

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.20, random\_state=0)

X\_train, X\_test, Y\_train= train\_test\_split(X, Y, test\_size=0.20, random\_state=0)

SubmittedCorrect!

Start project before checking your activities.

17

**Evaluation Metrics**

A model evaluation metric is a mechanism for assessing the performance of a machine learning model. A good metric should provide an informative summary of the error distribution.

*Choose the Right Evaluation Metric for this problem:*



Accuracy

MAE

Recall

RMSE

Precision

SubmittedCorrect!

Start project before checking your activities.

18.**Random Forest**

**Random forest**

Random Forest Regression is a supervised learning algorithm that uses an ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

Train a Random Forest with the following parameters: max\_depth= 3 and random\_state=42 and calculated the performance for the training set (keep only one significant value).

The RMSE is 174527.4

The RMSE is 173347

The RMSE is 174826

SubmittedCorrect!

Start project before checking your activities.

19.**Feature Selection**

**Feature Selection** is the method of reducing the input variable to any given model by using only relevant data and getting rid of noise in data. It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve.

* **Feature selection by Recursive Feature Elimination**: Recursive feature elimination (RFE) is a feature selection method that fits a model and removes the weakest feature (or features) until the specified number of features is reached.
* What are the features selected by RFEC, using a Random forest model with n\_estimators = 100 and random\_state = 0:

TOT\_LVG\_AREA', 'AGE', 'WATER\_DIST', 'CNTR\_DIST', 'SUBCNTR\_DI'

'TOT\_LVG\_AREA', 'OCEAN\_DIST', 'WATER\_DIST', 'CNTR\_DIST', 'SUBCNTR\_DI'

TOT\_LVG\_AREA', 'OCEAN\_DIST', 'WATER\_DIST', 'CNTR\_DIST'

SubmittedCorrect!

Start project before checking your activities.

#### Compare the performance of different models

In this task, the objective is to compare the performance of different algorithms and select the best one:

* Desicion Tree Regressor
* Bagging Regressor
* Random Forest Regressor
* Gradient Boosting Regressor
* XGBoost Regressor
* LGBM Regressor

Your task to complete:

* Create an empty dictionary to later use to store the metrics of each model.
* Complete the function below to train the models and store the result.

**def** **fit\_n\_print**(---, X\_train, ---, ----, y\_test):

model.fit(---, ---)

pred = model.predict(---)

r2 = metrics.r2\_score(y\_test, ---)

rmse = sqrt(metrics.mean\_squared\_error(---, ----))

scorer = {'r2' : metrics.make\_scorer(metrics.r2\_score),

'mse' : metrics.make\_scorer(metrics.mean\_squared\_error)

}

cv = cross\_validate(model, ----, ---, cv=10, scoring = ----)

r2\_cv = ---['test\_r2'].mean()

----- = np.----([sqrt(----) **for** mse **in** cv['test\_mse']])

**return** r2, rmse, r2\_cv, rmse\_cv, pred

20

**Best models performance**

The two models that present the best performance in term of the evaluation metrics (Highest R2 and lowest RMSE):

Random Forest

LGBM

Gradient Boosting

XGBoost

SubmittedCorrect!

Start project before checking your activities.