Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

1.25 to 1.5

Question Number : 181 Question Id : 640653577935 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Short Answer Question

What is the value of *b*? Enter your answer correct to two decimal places.

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

0.25 to 0.55

MLP

Section Id :	64065339077
Section Number :	12
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	25
Number of Questions to be attempted :	25
Section Marks :	50

Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065382619
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 182 Question Id : 640653577936 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MACHINE LEARNING PRACTICE (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?

CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531929925. ✔ YES

6406531929926. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	64065382620
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 183 Question Id : 640653577937 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following things can be observed from a Histogram ?

Options :

6406531929927. ✔ The range of scale of a numerical feature in the data

6406531929928. ✔ The distribution of a numerical feature in the data

6406531929929. ✖ The null values present in a feature of the data

6406531929930. ✖ The correlation between features and labels in the data

Question Number : 184 Question Id : 640653577945 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following are use cases of ColumnTransformer?

Options :

6406531929964. ✔ Data has some numerical and some categorical features.

6406531929965. ✖ Data has only categorical features and all of them are nominal.

6406531929966. ✔ Data has only categorical features, however, some features are ordinal and some are nominal.

6406531929967. ✖ Data has only numerical features, and all of them are uniformly distributed in range of 0 and 1 (both inclusive).

Question Number : 185 Question Id : 640653577946 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider a regression dataset, the features are "Temperature" and "Humidity", and the label is "Precipitation" (i.e. rain fall in centimeter), both the features are numerical and there are no missing values in the dataset. Following code snippet trains a simple model on this dataset, assume necessary imports:

```
data = pd.read_csv('dataset.csv')
X = data[data.columns[:-1]]
y = data[data.columns[-1]]
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size = 0.8)
mms = MinMaxScaler()
X_train['Temperature'] = mms.fit_transform(X_train['Temperature'])
X_train['Humidity'] = mms.fit_transform(X_train['Humidity'])


X_test['Temperature'] = mms.fit_transform(X_test['Temperature'])
X_test['Humidity'] = mms.fit_transform(X_test['Humidity'])

lr = LinearRegression().fit(X_train,y_train)
```

Choose the correct statements from the options:

Options :

6406531929968. ✖ The training set will have 20% of the data, which is a good practice.

6406531929969. ✔ The training set size is smaller than test set size.

6406531929970. ✔ The train and test samples are not scaled appropriately.

6406531929971. ✔ One of the fundamental assumption of Machine Learning, which is, training and test data belong to same distribution, is not upheld.

Question Number : 186 Question Id : 640653577948 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following options is/are correct?

Options :

6406531929976. ✔ If the data contains many outliers, scaling using the mean and variance of the data is likely to not work very well.

6406531929977. ✔ RFE first removes a few features which are not important and then fits and removes again and fits. It repeats this iteration until it reaches a suitable number of features.

6406531929978. ✖ A pipeline cannot have any feature selection steps.

6406531929979. ✔ If you will execute model.fit() for a second time, it will start training again using passed data and will remove the existing results.

Question Number : 187 Question Id : 640653577958 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Choose the correct option(s) for the below code

```
import numpy as np
import pandas as pd
from sklearn.datasets import fetch_california_housing
house = fetch_california_housing(as_frame=True)
```

Options :

6406531930021. ✖ **house.data.shape** will give the count of number of rows and columns for features(attributes) and labels(target) both

6406531930022. ✔ **house.target_names** will show the name of the target column

6406531930023. ✖ **house.data.tail()** will show first 5 rows of the data

6406531930024. ✔ **house.feature_names** will give a list of names of the columns of the feature matrix

Sub-Section Number : 3

Sub-Section Id : 64065382621

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 188 Question Id : 640653577938 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

For the below code which option will provide the samples(rows) containing outliers according to the standard **Boxplot** in the **weight** feature ?

```
dataset = {
    "height" : [100,103,102,102,176,150,143,133,122,200,230,222,143],
    "weight" : [30,32,33,34,33,33,37,48,44,51,100,123,111]
}
data = pd.DataFrame(dataset, columns=['height','weight'])
q1 = data['weight'].quantile(0.25)
q3 = data['weight'].quantile(0.75)
iqr = q3-q1
```

Options :

6406531929931. ✖ `data(data('weight') > (1.5*iqr + q3))`

6406531929932. ✖ `data[['weight'] > (1.5*iqr + q3)]`

6406531929933. ✔ `data[data['weight'] > (1.5*iqr + q3)]`

6406531929934. ✖ None of these

Question Number : 189 Question Id : 640653577939 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

What will be the output of the following code snippet ?

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder

data = {"fruits": ['apple','orange', 'banana', 'orange', 'apple'],
        "price": [10,20,5,20,10]}

df = pd.DataFrame(data)

ohe = OneHotEncoder()
print(ohe.fit_transform(df[['fruits']]).toarray())
```

Options :

6406531929935. ✖ [0 1 2 1 2]

6406531929936. ✖ [[0. 0. 0.],[1. 1. 1.],[2. 2. 2.],[1. 1. 1.],[2. 2. 2.]]

6406531929937. ✔ [[1. 0. 0.],[0. 0. 1.],[0. 1. 0.],[0. 0. 1.],[1. 0. 0.]]

6406531929938. ✖ This code snippet will throw an error

Question Number : 190 Question Id : 640653577940 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

What will be the output of the following code ?

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer

data = {"fruits": ['apple','orange', 'banana', 'orange', 'apple'],
        "price": [10,20,5,20,10],
        'color': ['red', 'orange', 'yellow', 'orange', 'red']}
df = pd.DataFrame(data)

transformers = [
    ('Ohe', OneHotEncoder(), [0,2]),
    ('scaler', StandardScaler(), [1])
]
ct = ColumnTransformer(transformers = transformers)

transformed_df = ct.fit_transform(df)
print(transformed_df.shape)
```

Options :

6406531929939. ✖ (7,5)

6406531929940. ✖ (3,5)

6406531929941. ✖ (5,3)

6406531929942. ✔ (5,7)

Question Number : 191 Question Id : 640653577941 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following is(are) true statements?

Statement 1: A dataset is splitted into the train set and the test set to obtain the better performance for the unseen data

statement 2 : We should not use the learning, observations and information gained from the test set while training the model

Options :

6406531929943. ✖ Statement 1 is True and statement 2 is False

6406531929944. ✖ Statement 1 is False and statement 2 is True

6406531929945. ✔ Statement 1 and statement 2 both are True

6406531929946. ✖ Both the statements are False

Question Number : 192 Question Id : 640653577944 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider following code snippet, assume necessary imports:

```
pipe = Pipeline([('reduce_dim', PCA()), ('clf', SVC())])
```

Which of the following represents correct method to tune hyper parameters with above pipeline object:

Options :

6406531929960. ✔
```
param_grid = dict(reduce_dim__n_components=[2, 5, 10],
                  clf__C=[0.1, 10, 100])
grid_search = GridSearchCV(pipe, param_grid=param_grid)
```

6406531929961. ✖
```
param_grid = dict(n_components__reduce_dim=[2, 5, 10],
                  C__clf=[0.1, 10, 100])
grid_search = GridSearchCV(pipe, param_grid=param_grid)
```

6406531929962. ✖
```
param_grid = dict(n_components=[2, 5, 10],
                  C=[0.1, 10, 100])
grid_search = GridSearchCV(pipe, param_grid=param_grid)
```

6406531929963. ✖
```
param_grid = dict(reduce_dim['n_components']=[2, 5, 10],
                  clf['C']=[0.1, 10, 100])
grid_search = GridSearchCV(pipe, param_grid=param_grid)
```

Question Number : 193 Question Id : 640653577950 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the following code:

```
from sklearn.datasets import make_regression
from sklearn.datasets import make_classification
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression

X_r, y_r = make_regression()
lr = LinearRegression()
lr.fit(X_r, y_r)
score1 = lr.score(X_r, y_r)

X_c, y_c = make_classification()
logr = LogisticRegression()
logr.fit(X_c, y_c)
score2 = logr.score(X_c, y_c)

print(score1)
print(score2)
```

Which metrics will be contained in score1 and score2 respectively?

Options :

6406531929986. ✖ Accuracy, Accuracy

6406531929987. ✖ R2 score, R2 score

6406531929988. ✖ Accuracy, R2 score

6406531929989. ✔ R2 score, Accuracy

6406531929990. ✖ F1 score, Precision

6406531929991. ✖ Precision, Recall

6406531929992. ✖ MAE, MSE

6406531929993. ✖ The code will result in an error

Question Number : 194 Question Id : 640653577953 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Suppose you have a trained stochastic gradient regressor model by enabling warm start parameter. What happens if you call the fit method again with the same model instance and different training data?

```
from sklearn.linear_model import SGDRegressor

model = SGDRegressor(warm_start=True)

X_train = [[0, 0], [1, 1]]
y_train = [0, 1]
model.fit(X_train, y_train)

# Call fit() again with different training data
X_train_new = [[2, 2], [3, 3]]
y_train_new = [2, 3]
model.fit(X_train_new, y_train_new)
```

Options :

6406531930001. ✖ The new training data is ignored, and the model continues training from the previously learned weights.

6406531930002. ✔ The new training data is used to update the model weights, but the previous weights are discarded.

6406531930003. ✖ An error is raised, indicating that the model has already been trained.

6406531930004. ✖ The model weights are reset, and the model begins training again from scratch.

Question Number : 195 Question Id : 640653577954 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

What is the purpose of the *tol* parameter in the fit method of the stochastic regressor?

```
from sklearn.linear_model import SGDRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

model = SGDRegressor()

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,
↪  random_state=42)

model.fit(X_train, y_train, early_stopping=True, validation_data=(X_val,
↪  y_val), validation_fraction=0.2, tol=0.001, n_iter_no_change=5)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
```

Options :

6406531930005. ✔ It specifies the tolerance level for early stopping based on the change in the validation error.

6406531930006. ✖ It controls the learning rate of the stochastic regressor during training.

6406531930007. ✖ It determines the maximum number of iterations for the training process.

6406531930008. ✖ It defines the fraction of the validation set used for early stopping.

Question Number : 196 Question Id : 640653577955 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

What will be the output of the following code?

```
from sklearn.datasets import make_regression
X, y = make_regression(n_samples = 8, n_features = 3)
from sklearn.preprocessing import PolynomialFeatures
poly_transform = PolynomialFeatures(degree=3, interaction_only=True)
X_trans = poly_transform.fit_transform(X,y)
print(X_trans.shape)
```

Options :

6406531930009. ✖ (8,4)

6406531930010. ✖ (8,5)

6406531930011. ✔ (8,8)

6406531930012. ✖ (8,10)

Question Number : 197 Question Id : 640653577956 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

How many models with different combinations of parameter values will get trained in the following code?

```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import SGDRegressor

params = [
    {'alpha': [0.001,0.01,0.1,1],'learning_rate': ['constant','optimal']},
    {'warm_start':[True], 'alpha':
     ↪  [0.0001,0.001],'learning_rate':['constant','invscaling']}]

grid= GridSearchCV(estimator= SGDRegressor(),
                   param_grid = params,
                   cv= 2,
                   scoring = 'neg_mean_squared_error',
                   return_train_score=True
                   )

grid.fit(X_train,y_train)
```

Options :

6406531930013. ✖ 10

6406531930014. ✔ 12

6406531930015. ✖ 14

6406531930016. ✖ 16

Question Number : 198 Question Id : 640653577957 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following will be the correct option for the below code to control the regularization rate ?

```
from sklearn.linear_model import Ridge
import numpy as np
model = Ridge(__________ = 0.001)
model.fit(X_train,y_train)
```

Options :

6406531930017. ✖ max_iter

6406531930018. ✖ lambda

6406531930019. ✔ alpha

6406531930020. ✖ solver

Question Number : 199 Question Id : 640653577959 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

To see the distribution of the data along each numerical feature which of the following codes will show all the histograms?

• Variable name **df** contains all the data as pandas.core.frame.DataFrame type

• Assume all the necessary imports are made

Options :

6406531930025. ✔ df.hist()

6406531930026. ✖ df.histplot()

6406531930027. ✖ pandas.hist(df)

6406531930028. ✖ seaborn.histogram(df)

Question Number : 200 Question Id : 640653577960 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following can affect performance of the Simple linear Regression while training the model?

Options :

6406531930029. ✔ Scaling

6406531930030. ✖ Kernel

6406531930031. ✖ Range of the target column

6406531930032. ✖ r2_score

Sub-Section Number : 4

Sub-Section Id : 64065382622

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 201 Question Id : 640653577942 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

which of the following methods come under sklearn's `model_selection` module ?

Options :

6406531929947. ✔ `train_test_split`

6406531929948. ✖ ColumnTransformer

6406531929949. ✔ GridSearchCV

6406531929950. ✖ Pipeline

6406531929951. ✔ StratifiedShuffleSplit

6406531929952. ✔ KFold

6406531929953. ✖ `mean_absolute_error`

6406531929954. ✔ `cross_val_score`

6406531929955. ✖ trees

Question Number : 202 Question Id : 640653577949 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following code block:

```
from sklearn.linear_model import linear_regression
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import ShuffleSplit
lin_reg = linear_regression()
shuffle_split = ShuffleSplit(n_splits=5, test_size=0.2, random_state=42)
score = cross_val_score(lin_reg, X, y, cv=shuffle_split,
scoring='----------------------')
```

Which of the following may be appropriate to be filled in the blank space?

Options :

6406531929980. ✖ mean_squared_error

6406531929981. ✔ neg_mean_squared_error

6406531929982. ✔ r2

6406531929983. ✖ neg_r2

6406531929984. ✖ accuracy

6406531929985. ✖ neg_accuracy

Sub-Section Number : 5

Sub-Section Id : 64065382623

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 203 Question Id : 640653577943 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following are correct statements?

Options :

6406531929956. ✔ Data with missing values can not be used to train a logistic or linear regression model.

6406531929957. ✖ KNN model is agnostic to scale of numerical features.

6406531929958. ✔ Like KNN imputer, other models e.g. decision tree, can also be used for imputation.

6406531929959. ✖ Filling 0 (zero) in place of missing values, is always the best approach for imputation.

Sub-Section Number : 6

Sub-Section Id : 64065382624

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 204 Question Id : 640653577947 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1

Question Label : Multiple Choice Question

What will be the output of the following code:

```
from sklearn.feature_extraction import DictVectorizer

X = [{'feature_1': 3, 'feature_2': 1}, {'feature_1': 2, 'feature_3': 3}]
transformer = DictVectorizer(sparse= False)
print(transformer.fit_transform(X))
```

Options :

6406531929972. ✖
```
[[3. 1.]
 [2. 3.]]
```

6406531929973. ✖
```
[[3. 1.]
 [3. 0.]]
```

6406531929974. ✔
```
[[3. 1. 0.]
 [2. 0. 3.]]
```

6406531929975. ✖
```
[[3. 1. 0.]
 [2. 3. 0.]]
```

Sub-Section Number :	7
Sub-Section Id :	64065382625
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 205 Question Id : 640653577951 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following code:

```
import numpy as np
from sklearn.model_selection import ShuffleSplit
X = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [1, 3], [2, 3], [3, 3], [4, 3]])
y = np.array([0, 1, 0, 1, 1, 0, 1, 0])
rs = ShuffleSplit(n_splits=5, test_size=.25, random_state=0)
for each in rs.split(X):
  print(each[0], each[1])
```

Which of the following may be the correct output of the above code?:

Options :

6406531929994. ✔

[1 7 3 0 5 4] [6 2]
[3 7 0 4 2 5] [1 6]
[3 4 7 0 6 1] [5 2]
[6 7 3 4 1 0] [2 5]
[1 6 3 2 0 7] [4 5]

6406531929995. ✖

[1 7 3 0 5] [4 6 2]
[3 7 0 4 2] [5 1 6]
[3 4 7 0 6] [1 5 2]
[6 7 3 4 1] [0 2 5]
[1 6 3 2 0] [7 4 5]

6406531929996. ✖

[1 7 3 0] [5 4 6 2]
[3 7 0 4] [2 5 1 6]
[3 4 7 0] [6 1 5 2]
[6 7 3 4] [1 0 2 5]
[1 6 3 2] [0 7 4 5]

6406531929997. ✖

[1 7 3 0 5] [5 6 2]
[3 7 0 4 2] [4 1 6]
[3 4 7 0 6] [7 5 2]
[6 7 3 4 1] [6 2 5]
[1 6 3 2 0] [1 4 5]

Question Number : 206 Question Id : 640653577952 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following code:

```
import numpy as np
from sklearn.linear_model import LinearRegression
X = np.array([[1, 1], [1, 2], [2, 2], [2, 3], [2, 1], [3, 3]])
# y = 1 * x_0 + 2 * x_1 + 3
y = np.dot(X, np.array([1, 2])) + 3

reg1 = LinearRegression(fit_intercept = False).fit(X, y)
s1 = reg1.score(X, y)

reg2 = LinearRegression(fit_intercept = True).fit(X, y)
s2 = reg2.score(X, y)
```

Which of the following is more likely to be true?

Options :

6406531929998. ✖ s1 = s2

6406531929999. ✔ s1 < s2

6406531930000. ✖ s1 > s2

BDM

Section Id :	64065339078
Section Number :	13
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	11
Number of Questions to be attempted :	11
Section Marks :	16
Display Number Panel :	Yes
Group All Questions :	No