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

Correct Marks : 3

Question Label : Short Answer Question

What will be the predicted label according to the Naive Bayes decision rule?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

1

MLP

Section Id :	64065348511
Section Number :	13
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	24
Number of Questions to be attempted :	24
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 :	640653100866
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 193 Question Id : 640653689612 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 :

6406532306945. ✔ YES

6406532306946. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	640653100867
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 194 Question Id : 640653689626 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

You are working with a dataset containing 1000 samples, aiming to classify them using the KNeighborsClassifier from scikit-learn. After trying an initial configuration, you observe that the model seems to be overfitting, with the following accuracies:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Initial Configuration
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
train_acc = accuracy_score(y_train, knn.predict(X_train))
val_acc = accuracy_score(y_val, knn.predict(X_val))
```

- Training accuracy: 98%
- Validation accuracy: 65%

After observing such performance of the model, Which of the following values for n_neighbors would be more suitable to try next?

Options :

6406532306996. ✱ 1

6406532306997. ✱ 2

6406532306998. ✔ 10

6406532306999. ✱ 500

Question Number : 195 Question Id : 640653689630 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

Consider the following code segment which uses CountVectorizer on a set of documents:

```
from sklearn.feature_extraction.text import CountVectorizer

documents = [
    'apple orange banana',
    'apple apple',
    'banana orange',
    'apple banana orange orange'
]

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(documents)
```

After executing the code, what will be the shape of matrix X?

Options :

6406532307012. ✔ (4, 3)

6406532307013. ✖ (3, 4)

6406532307014. ✖ (4, 4)

6406532307015. ✖ (3, 3)

Sub-Section Number :   3

Sub-Section Id :   640653100868

Question Shuffling Allowed :   Yes

Is Section Default? :   null

Question Number : 196 Question Id : 640653689613 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

Assume train data (X_train, y_train) and test data (X_test) is given as numpy array and you build and train a LogisticRegression model. Which of the following options might possibly be the predicted class of first two samples(rows) of the test data according to the code given below?

```
>>> from sklearn.linear_model import LogisticRegression
>>> log_reg = LogisticRegression()
>>> log_reg.fit(X_train,y_train)

>>> print(log_reg.classes_)
[0,1,2] #output of above code

>>> print(log_reg.predict_proba(X_test[[0]]))
[[2.73e-45, 1.21e-51, 1.00e+00]] #output of above code

>>> print(log_reg.predict_proba(X_test[[1]]))
[[7.09e-29, 1.00e+00, 2.02e-36]] #output of above code

>>> print(log_reg.predict(X_test[0:2]))
```

Options :

6406532306947. ✖ $[2.73 , 1.00]$

6406532306948. ✖ $[\text{True} , \text{False}]$

6406532306949. ✔ $[2 , 1]$

6406532306950. ✖ cannot be found

Question Number : 197 Question Id : 640653689614 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 block of code given below:

```python
from sklearn.metrics import confusion_matrix
y_true = ["cat", "ant", "cat", "cat", "ant", "bird"]
y_pred = ["ant", "ant", "cat", "cat", "ant", "cat"]
cm = confusion_matrix(y_true, y_pred, labels=["ant", "bird", "cat"])
print(cm)
```

Which of the following option represents the print output :

Options :

6406532306951. ✔
```
[[2, 0, 0],
 [0, 0, 1],
 [1, 0, 2]]
```

6406532306952. ✖
```
[[1, 0, 2],
 [2, 0, 0],
 [0, 0, 1]]
```

6406532306953. ✖
```
[[1, 0, 2],
 [0, 0, 1],
 [2, 0, 0]]
```

6406532306954. ✖
```
[[2, 0, 0],
 [1, 0, 0],
 [1, 0, 2]]
```

Question Number : 198 Question Id : 640653689618 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

Choose the correct output of the following code?

```
data = [[1, 3],
        [2, 4],
        [3, 5]]
from sklearn.preprocessing import PolynomialFeatures
pf = PolynomialFeatures(degree=3,interaction_only=True)
print(pf.fit_transform(data))
```

Options :

6406532306961. ✖
```
[[1, 2, 3,  2,  4],
 [1, 2, 4,  6,  8],
 [1, 2, 5, 10, 12]]
```

6406532306962. ✔
```
[[1, 1, 3,  3],
 [1, 2, 4,  8],
 [1, 3, 5, 15]]
```

6406532306963. ✖
```
[[1, 1, 3, 1,  3,  9,  1,  3,  9,  27],
 [1, 2, 4, 4,  8, 16,  8, 16, 32,  64],
 [1, 3, 5, 9, 15, 25, 27, 45, 75, 125]]
```

6406532306964. ✖
```
[[1, 1, 2, 2,  3,  3,  4,  4],
 [1, 1, 4, 4,  9,  9, 16, 16]]
 [1, 1, 9, 9, 27, 27, 64, 64]
```

Question Number : 199 Question Id : 640653689621 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 following code implement?

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X,y)
```

where X and y are the training data.

Options :

6406532306974. ✳ It will perform regression on the given data.

6406532306975. ✳ It will generate synthetic regression data.

6406532306976. ✔ It will perform classification on the given data.

6406532306977. ✳ It will generate synthetic classification data.

Question Number : 200 Question Id : 640653689623 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 method of classification needs more than n classifiers, where n is the number of classes?

Options :

6406532306982. ✳ OneVsRestClassifier

6406532306983. ✔ OneVsOneClassifier

6406532306984. ✳ OutputCodeClassifier

6406532306985. ✳ MultiOutputClassifier

Question Number : 201 Question Id : 640653689624 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:

```
estimator = RidgeClassifier(normalize=False, _____='auto')
pipe_ridge = make_pipeline(MinMaxScaler(),estimator)
pipe_ridge.fit(x,y)
```

If we want to apply the ridge classifier on X and choose the appropriate algorithm to train on the data, what will be the missing attribute?

Options :

6406532306986. ✷ alpha

6406532306987. ✷ tol

6406532306988. ✔ solver

6406532306989. ✷ learner

6406532306990. ✷ algo

6406532306991. ✷ algorithm

Question Number : 202 Question Id : 640653689627 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.neighbors import KNeighborsClassifier
import numpy as np

X_train = np.array([[1, 0.5],   [2, 1], [3, 1.5],   [4, 2], [5, 2.5],
                    [6, 3], [7, 3.5],   [8, 4], [9, 4.5],   [10, 5]])

y_train = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2]

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)
```

Given a single test data point X_test, what will be the output of the following code?

```
print(knn.predict(X_test))
```

Options :

6406532307000. ✶ 0

6406532307001. ✶ 1

6406532307002. ✔ 2

6406532307003. ✶ Can not be determined without knowing the test data point.

Question Number : 203 Question Id : 640653689629 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 statements accurately describes the difference between SVC and LinearSVC in scikit-learn?

Options :

6406532307008. ✶ SVC supports only non-linear kernels while LinearSVC supports only linear kernels.

6406532307009. ✔ LinearSVC is designed specifically for linear SVM problems and does not support the use of kernels, while SVC supports both linear and non-linear kernel functions.

6406532307010. ✶ Both SVC and LinearSVC are optimized for non-linear problems and make use of kernel functions.

6406532307011. ✶ LinearSVC is a regression model while SVC is a classification model.

Sub-Section Number :   4

Sub-Section Id :   640653100869

Question Shuffling Allowed :   Yes

Question Number : 204 Question Id : 640653689620 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 classifier and select the correct option.

```
estimator = SGDClassifier(loss='log',
                          penalty='l2',
                          max_iter=1,
                          warm_start=True,
                          eta0=0.01,
                          alpha=0,
                          learning_rate='constant',
                          random_state=1729)
```

Options :

6406532306970. ✖ It applies the perceptron classification with regularization.

6406532306971. ✖ It applies the perceptron classification without regularization.

6406532306972. ✖ It applies the logistic regression with regularization.

6406532306973. ✔ It applies the logistic regression without regularization.

Question Number : 205 Question Id : 640653689634 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 below code for a given training data:

```python
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestClassifier


param_distributions = {"n_estimators" : range(3,100,2),
                       "max_depth": range(3,40,2),
                       "min_samples_split" : [3,4,5,6,7]}


RS_CV = RandomizedSearchCV(estimator=RandomForestClassifier(random_state=0),
            param_distributions=param_distributions,
            cv=3,
            n_iter=12)

RS_CV.fit(X_train,y_train)
```

Which of the following option(s) are True ?

Options :

6406532307029. ✔ A total of 12 estimators will be trained, with each estimator using 3-fold crossvalidation

6406532307030. ✖ The parameter combination will be the same in every run because random_state is set to 0.

6406532307031. ✖ Given code will throw an error because all the parameters are not presented as a list

6406532307032. ✖ All of the options are incorrect

Sub-Section Number :   5

Sub-Section Id :   640653100870

Question Shuffling Allowed :   Yes

Is Section Default? :   null

Question Number : 206 Question Id : 640653689632 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

According to DecisionTreeClassifier parameters which of the following option will have least fitting(underfit) for the same data.

Options :

6406532307020. ✖
```
DecisionTreeClassifier(max_depth = None,
                       min_samples_split= 20,
                       min_samples_leaf = 10)
```

6406532307021. ✖
```
DecisionTreeClassifier(max_depth = 20,
                       min_samples_split= 15,
                       min_samples_leaf = 8)
```

6406532307022. ✔
```
DecisionTreeClassifier(max_depth = 2,
                       min_samples_split= 30,
                       min_samples_leaf = 15)
```

6406532307023. ✖
```
DecisionTreeClassifier(max_depth = 5,
                       min_sample_split= 18,
                       min_sample_leaf = 12)
```

Sub-Section Number :	6
Sub-Section Id :	640653100871
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 207 Question Id : 640653689616 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

For the given X_train (in pandas DataFrame) below which of the following options can successfully impute the null values ?

	Weight	Education
0	NaN	School
1	56.0	High-School
2	45.0	Bachelor
3	NaN	Masters
4	40.0	School
5	40.0	High-School
6	20.0	Bachelor
7	67.0	NaN
8	20.0	School
9	35.0	NaN

Options :

```python
from sklearn.pipeline import Pipeline
pipe = Pipeline( [
    ("weight_si", SimpleImputer(strategy="median")),
    ("education_si", SimpleImputer(strategy="most_frequent"))
    ])

X_train = pipe.fit_transform(X_train)
```
6406532306956. ✖

```python
from sklearn.pipeline import FeatureUnion
union = FeatureUnion( [
    ("weight_si", SimpleImputer(strategy="median")),
    ("education_si", SimpleImputer(strategy="most_frequent"))
    ])
```
6406532306957. ✖   `X_train = union.fit_transform(X_train)`

```python
from sklearn.impute import SimpleImputer

weight_si = SimpleImputer(strategy="median")
X_train['Weight'] = weight_si.fit_transform(X_train[['Weight']])

education_si = SimpleImputer(strategy="most_frequent")
X_train['Education']=education_si.fit_transform(X_train[['Education']])
```
6406532306958. ✔

6406532306959. ✔

```
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer

ct= ColumnTransformer(transformers= [
("weight_si",SimpleImputer(strategy="median"), ["Weight"]),
("education_si",SimpleImputer(strategy="most_frequent"),["Education"])
])

X_train= ct.fit_transform(X_train)
```

Question Number : 208 Question Id : 640653689619 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 is/are true about DummyClassifier?

Options :

6406532306965. ✔ DummyClassifier makes predictions that ignore the input features.

6406532306966. ✔ DummyClassifier serves as a simple baseline to compare against other more complex classifiers.

6406532306967. ✔ The predictions of DummyClassifier typically depend on values observed in the y parameter passed to fit().

6406532306968. ✖ The predictions of DummyClassifier typically depend on values observed in the X parameter passed to fit().

6406532306969. ✖ All of these.

Question Number : 209 Question Id : 640653689622 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 solver in LogisticRegression() is a better choice for a large dataset?

Options :

6406532306978. ✔ sag

6406532306979. ✔ saga

6406532306980. ✖ lbfgs

6406532306981. ✖ liblinear

Question Number : 210 Question Id : 640653689625 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

You are analyzing a dataset with features X_{train} and targets y_{train}. After standardizing the feature set, you decide to apply the KNeighborsClassifier from scikit-learn to classify data points. You use the following code:

```
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n_neighbors=5,
                             weights='distance',
                             metric='minkowski', p=1)

model.fit(X_train, y_train)
```

Given the above code configuration for KNeighborsClassifier, which of the following statements are true? (Select all that apply)

Options :

6406532306992. ✖ The classifier is using the Euclidean distance metric.

6406532306993. ✖ Outliers will have a stronger influence on predictions.

The classifier is using the Manhattan distance metric and will calculate

6406532306994. ✔ distance as the sum of absolute differences for each feature.

When `weights` is set to `distance`, points closer to the decision boundary will

6406532306995. ✔ have a stronger influence on predictions than those farther away.

Question Number : 211 Question Id : 640653689628 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 the following Python code where you are using the SVC classifier to categorize data from a binary classification problem:

```python
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline

clf = make_pipeline(StandardScaler(),
                    SVC(C=1.0, kernel='rbf', gamma='scale'))

clf.fit(X_train, y_train)
prediction = clf.predict(X_test)
```

Assume that X_train, y_train, and X_test are training feature matrix, label vector, and test feature matrix, respectively. Which of the following statements is/are true using the code given above?

Options :

The model doesn't need scaling because SVMs are not sensitive to feature

6406532307004. ✖ scales.

Using a pipeline that incorporates `StandardScaler` before `SVC`. This ensures

6406532307005. ✔ that each feature is scaled before fitting the model.

6406532307006. ✔ The C parameter controls the trade-off between achieving a low error on the training data and maximizing the margin.

6406532307007. ✖ The model is using a linear kernel due to the nature of the dataset.

Question Number : 212 Question Id : 640653689635 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 advantages of using ensemble methods in machine learning?

Options :

6406532307033. ✔ Improved model performance

6406532307034. ✔ Reduced overfitting

6406532307035. ✖ Faster model training

6406532307036. ✖ Simplicity of model interpretation

Sub-Section Number :	7
Sub-Section Id :	640653100872
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 213 Question Id : 640653689631 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 option(s) are True ?

Options :

6406532307016. ✖ ccp_alpha parameter helps in post-prunning and hence classifier will prioritize it more than pre-prunning parameters.

6406532307017. ✔ In DecisionTreeClassier if a sample (data point) meets the condition at the parent node (i.e., if it satisfies the split criteria), it goes to the left child node; Otherwise, it goes to the right child node.

6406532307018. ✖ RandomForestClassifier is better than DecisionTreeClassifer because it's Loss function convergences much faster while optimizing through gradient descent.

6406532307019. ✔ In DecisionTreeClassifier there is no predefined priority in parameters. The tree will stop growing as soon as the first parameter condition is met.

Question Number : 214 Question Id : 640653689633 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

Suppose In a classification problem you want to use BaggingClassifier, which of the following estimator(s) could be used as base estimator in that?

Options :

6406532307024. ✔ tree.DecisionTreeClassifier()

6406532307025. ✖ svm.SVR()

6406532307026. ✔ linear_model.Perceptron()

6406532307027. ✖ cluster.KMeans()

Sub-Section Number :   8

Sub-Section Id :   640653100873

Question Shuffling Allowed :   Yes

Is Section Default? :   null

Question Number : 215 Question Id : 640653689615 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

Consider given below confusion matrix code :

```
from sklearn.metrics import confusion_matrix
y_true = ["cat", "ant", "cat", "cat", "ant", "bird"]
y_pred = ["ant", "ant", "cat", "cat", "ant", "cat"]
cm = confusion_matrix(y_true, y_pred, labels=["ant", "bird", "cat"])
```

Determine the recall score for class "ant" in the given confusion_matrix?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

1

Question Number : 216 Question Id : 640653689617 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 will be the output of the following code:

```
from sklearn.preprocessing import MaxAbsScaler
a = [[-3],[ 0],[-2],[ 2],[-1],[-4]]
mas = MaxAbsScaler()
scaled_a = mas.fit_transform(a)
print(scaled_a.max())
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

0.5

Business Analytics

Section Id :	64065348512
Section Number :	14
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	7
Number of Questions to be attempted :	7
Section Marks :	20
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 :	640653100874
Question Shuffling Allowed :	No