MLP

Section Id :	64065360930
Section Number :	8
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	23
Number of Questions to be attempted :	23
Section Marks :	50
Display Number Panel :	Yes
Section Negative Marks :	0
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	No
Section Maximum Duration :	0
Section Minimum Duration :	0
Section Time In :	Minutes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653126865
Question Shuffling Allowed :	No

Question Number : 122 Question Id : 640653852456 Question Type : MCQ

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 :

6406532867227. ✔ YES

6406532867228. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	640653126866
Question Shuffling Allowed :	Yes

Question Number : 123 Question Id : 640653852457 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Given the following code for **Polynomial Regression**:

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import numpy as np

X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
y = np.array([1, 4, 9, 16, 25])

poly = PolynomialFeatures(degree=2,interaction_only=False)
X_poly = poly.fit_transform(X)
model = LinearRegression()
model.fit(X_poly, y)

print(model.coef_)
```

What are the coefficients of the model?

Options :

6406532867229. ✖ [0, 1, 1]

6406532867230. ✖ [0, 2, 1]

6406532867231. ✔ [0, 0, 1]

6406532867232. ✖ [0, 0, 2]

Question Number : 124 Question Id : 640653852470 Question Type : MCQ
Correct Marks : 3

Question Label : Multiple Choice Question

What is 'naive' assumption in classifiers based on Naive Bayes?

Options :

6406532867279. ✖ All the classes are conditionally independent of each other.

6406532867280. ✖ All the classes are conditionally dependent on each other.

6406532867281. ✖ All the features are conditionally dependent on each other.

6406532867282. ✔ All the features are conditionally independent of each other.

Question Number : 125 Question Id : 640653852471 Question Type : MCQ
Correct Marks : 3

Question Label : Multiple Choice Question

Consider following code snippet:

```
estimator = RidgeClassifier(normalize=False, _ _ _ _ _=0)
pipe_ridge = make_pipeline(MinMaxScaler(),estimator)
pipe_ridge.fit(x,y)
```

If we want to apply the ridge classifier on X with no regularization, what will be the missing attribute.

Options :

6406532867283. ✖ cv

6406532867284. ✖ reg_rate

6406532867285. ✔ alpha

6406532867286. ✖ tol

6406532867287. ✖ lambda

6406532867288. ✖ None of these

Sub-Section Number : 3

Sub-Section Id : 640653126867

Question Shuffling Allowed : Yes

Question Number : 126 Question Id : 640653852458 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

What is the effect of increasing the regularization parameter alpha in **Ridge Regression**?

Options :

6406532867233. ✖ It increases the complexity of the model.

6406532867234. ✔ It reduces the complexity of the model.

6406532867235. ✖ It has no effect on the model.

6406532867236. ✖ It increases the model's sensitivity to the training data.

Question Number : 127 Question Id : 640653852459 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

Given the code below:

```
from sklearn.datasets import fetch_california_housing
from sklearn.linear_model import LinearRegression

data = fetch_california_housing()
X, y = data.data, data.target

model = LinearRegression()
model.fit(X, y)

print(model.coef_)
```

What do the coefficients represents?

Options :

6406532867237. ✔ The importance of each feature in predicting the target.

6406532867238. ✖ The residuals of the model.

6406532867239. ✖ The intercept of the regression line.

6406532867240. ✖ It represents the values of hyperparameters of the model.

Question Number : 128 Question Id : 640653852460 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

How can we use both **Ridge** and **Lasso** Regularization in a machine learning model?

Options :

6406532867241. ✖ By setting **penalty** parameter to **l2**

6406532867242. ✖ By setting **penalty** parameter to **l1**

6406532867243. ✔ By setting **penalty** parameter to **elasticnet**

6406532867244. ✖ By setting **penalty** parameter to **l3**

Question Number : 129 Question Id : 640653852461 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following types of classification problems is being solved when a model predicts multiple labels(greater than 2) for each instance?

Options :

6406532867245. ✖ Binary class, single label classification.

6406532867246. ✔ Multi class, multi label classification.

6406532867247. ✖ Binary class, multi label classification.

6406532867248. ✖ Multi class, single label classification.

Question Number : 130 Question Id : 640653852462 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

Given the code snippet and assume if any necessary requirements:

```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
param_grid = {'C': [0.1, 1, 10],
              'penalty': ['l1', 'l2'],
              'solver': ['liblinear']}
clf = GridSearchCV(LogisticRegression(), param_grid, cv=5)
clf.fit(X_train, y_train)
print(clf.best_params_)
```

What does cv=5 signify in this context?

Options :

6406532867249. ✖ The training data is split into 5 different datasets, each used to train a separate model.

6406532867250. ✔ The model is trained and evaluated using 5-fold cross-validation for hyperparameter tuning

6406532867251. ✖ Five different models are trained on 5 resampled datasets of training data.

6406532867252. ✖ The dataset is divided into 5 parts and trained in a single pass without cross-validation.

Question Number : 131 Question Id : 640653852463 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the following code and assume all the imports being made:

```
from sklearn.linear_model import Perceptron

X_train, X_test, y_train, y_test = MNIST()
clf = Perceptron()
clf.fit(X_train, y_train)

print(clf.score(X_test, y_test))
```

What does the score method compute in this context?

Options :

6406532867253. ✖ The number of correctly classified samples.

6406532867254. ✔ The accuracy of the classifier.

6406532867255. ✖ The loss function value.

6406532867256. ✖ The number of misclassified samples.

Question Number : 132 Question Id : 640653852464 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

Given the following code snippet for a **multi-label** classification problem:

```
from sklearn.multioutput import MultiOutputClassifier
from sklearn.linear_model import LogisticRegression

model = MultiOutputClassifier(LogisticRegression())
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

What does **MultiOutputClassifier** do in this context?

Options :

6406532867257. ✖ Trains a single **Logistic Regression** model for all labels.

6406532867258. ✖ Combines **Logistic Regression** with another model.

6406532867259. ✔ Trains a separate **Logistic Regression** model for each label.

6406532867260. ✖ Applies **Logistic Regression** in a one-vs-rest strategy for multi-class classification.

Question Number : 133 Question Id : 640653852465 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

What is the main advantage of using **RandomizedSearchCV** over **GridSearchCV** ?

Options :

6406532867261. ✖ **RandomizedSearchCV** is always faster.

6406532867262. ✖ **RandomizedSearchCV** evaluates all possible combinations of hyperparameters.

6406532867263. ✔ **RandomizedSearchCV** can sample a larger hyperparameter space with fewer iterations.

6406532867264. ✖ **RandomizedSearchCV** always finds the best model.

Question Number : 134 Question Id : 640653852467 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

When using the `SGDClassifier` with `loss='squared_loss'` and `penalty='l2'` which of the model given in option are you training?

Options :

6406532867266. ✖ LogisticRegression

6406532867267. ✖ Perceptron

6406532867268. ✔ RidgeClassifier

6406532867269. ✖ Support Vector Machine (SVM)

Question Number : 135 Question Id : 640653852468 Question Type : MCQ
Correct Marks : 2
Question Label : Multiple Choice Question
What type of classification problem is the **Perceptron** algorithm best suited for?
Options :
6406532867270. ✔ Linearly separable binary classification problems.
6406532867271. ✖ Non-linear classification problems.
6406532867272. ✖ Regression problems.
6406532867273. ✖ Unsupervised Learning Problems

Question Number : 136 Question Id : 640653852469 Question Type : MCQ
Correct Marks : 2
Question Label : Multiple Choice Question
What will be the output of the following code? Given that output of iris.target_names is ['setosa', 'versicolor', 'virginica']

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
iris = load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.4,
                                                    random_state=1)
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
gnb.classes_
```

Options :
6406532867274. ✔ [0, 1, 2]
6406532867275. ✖ [0, 1]
6406532867276. ✖ ['setosa', 'versicolor', 'virginica']
6406532867277. ✖ ['setosa', 'versicolor"]
6406532867278. ✖ None of these

Question Number : 137 Question Id : 640653852472 Question Type : MCQ
Correct Marks : 2
Question Label : Multiple Choice Question
What is the default loss value in *SGDClassifier* API and it gives which classifier?
Options :

6406532867289. ✖ 'log loss', LogisticRegressor

6406532867290. ✖ 'log loss', LogisticClassifier

6406532867291. ✔ 'hinge', SVM

6406532867292. ✖ 'hinge', Perceptron

Question Number : 138 Question Id : 640653852477 Question Type : MCQ
Correct Marks : 2

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=4)
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 most suitable to try next?

Options :

6406532867304. ✖ 1

6406532867305. ✖ 2

6406532867306. ✔ 10

6406532867307. ✖ 500

Sub-Section Number : 4

Sub-Section Id : 640653126868

Question Shuffling Allowed : Yes

Question Number : 139 Question Id : 640653852466 Question Type : SA
Correct Marks : 3

Question Label : Short Answer Question

Calculate the precision score for the class "dog" in the following code:

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

Response Type : Numeric
Evaluation Required For SA : Yes
Show Word Count : Yes
Answers Type : Equal
Text Areas : PlainText
Possible Answers :
0.67

Question Number : 140 Question Id : 640653852476 Question Type : SA
Correct Marks : 3
Question Label : Short Answer Question

What will be the output of the following code ?

```
import numpy as np
from sklearn.neighbors import KNeighborsClassifier

X_train = np.array([[1,100],[4,400],[5,500],[6,600],[8,800],[9,900],
                    [11,1100],[12,1200],[15,1500], [18,1800],[19,1900]])

y_train = np.array([0,0,1,1,1,2,2,2,2,2,2])

X_test = np.array([[2,200]])

knn = KNeighborsClassifier(n_neighbors= len(y_train),
                           metric="euclidean",
                           weights= 'uniform')
knn.fit(X_train,y_train)
print(knn.predict(X_test))
```

Response Type : Numeric
Evaluation Required For SA : Yes
Show Word Count : Yes
Answers Type : Equal
Text Areas : PlainText
Possible Answers :
2

Sub-Section Number : 5
Sub-Section Id : 640653126869
Question Shuffling Allowed : Yes

Question Number : 141 Question Id : 640653852475 Question Type : SA
Correct Marks : 2
Question Label : Short Answer Question

What will be the output of the following code?

```
import numpy as np
from sklearn.impute import KNNImputer
X = np.array([[5,6,3],[np.nan,1,5],[0,2,8],[4,4,2]])
knn = KNNImputer(n_neighbors=2,weights="uniform")
X_trf= knn.fit_transform(X)
print(X_trf[1][0])
```

Response Type : Numeric
Evaluation Required For SA : Yes
Show Word Count : Yes
Answers Type : Equal
Text Areas : PlainText
Possible Answers :
2

Sub-Section Number :	6
Sub-Section Id :	640653126870
Question Shuffling Allowed :	Yes

Question Number : 142 Question Id : 640653852473 Question Type : MSQ
Correct Marks : 2 Max. Selectable Options : 0
Question Label : Multiple Select Question

Fill in the missing parameter value in the following estimator that can be used to classify the data

```
from sklearn.svm import SVC
clf = SVC(kernel = ______)
clf.fit(X, y)
```

Options :
6406532867293. ✔ 'poly',
6406532867294. ✖ 'lasso'
6406532867295. ✔ 'rbf',
6406532867296. ✖ 'scale'

Question Number : 143 Question Id : 640653852478 Question Type : MSQ
Correct Marks : 2 Max. Selectable Options : 0
Question Label : Multiple Select Question
Which of the following **scikit-learn** model supports incremental learning through partial_fit

method ?

Options :

6406532867308. ✔ SGDClassifier

6406532867309. ✖ LinearRegression

6406532867310. ✔ Perceptron

6406532867311. ✖ SVC

Sub-Section Number :	7
Sub-Section Id :	640653126871
Question Shuffling Allowed :	Yes

Question Number : 144 Question Id : 640653852474 Question Type : MSQ

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following options are true for regularization parameter C in sklearn.svm.SVC ?

Options :

6406532867297. ✔ Large value of the regularization parameter C will overfit the training set and complex decision boundaries will form.

6406532867298. ✖ Large value of the regularization parameter C will underfit the training set and smooth decision boundaries will form.

6406532867299. ✖ Small value of the regularization parameter C will overfit the training set and complex decision boundaries will form.

6406532867300. ✔ Small value of the regularization parameter C will underfit the training set and smooth decision boundaries will form.

6406532867301. ✖ None of these

MLT

Section Id :	64065360931
Section Number :	9
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	11
Number of Questions to be attempted :	11
Section Marks :	40
Display Number Panel :	Yes
Section Negative Marks :	0
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	No
Section Maximum Duration :	0
Section Minimum Duration :	0