

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

5

## MLP

Number of Questions : 22

Section Marks : 50

Question Number : 77 Question Type : MCQ

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL: MACHINE LEARNING PRACTICES"

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 :

A. ✓ YES

B. ✗ NO

Question Number : 78 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

Which of the following will create a confusion matrix plot from  $y$  and  $y_{\hat{}}$ ?

**Options :**

A. ✓ `cm_display = ConfusionMatrixDisplay.from_predictions(y,y_hat_)`  
`plt.show()`

B. ✗ `cm_display = ConfusionMatrixDisplay(y,y_hat_)`  
`plt.show()`

C. ✗ `cm_display = ConfusionMatrix.from_predictions(y,y_hat_)`  
`plt.show()`

D. ✗ `cm_display = ConfusionMatrix(y,y_hat_)`  
`plt.show()`

**Question Number : 79 Question Type : MCQ**

**Correct Marks : 2**

Question Label : Multiple Choice Question

We have a dataset which has only numerical features. We are implementing the following model on this dataset. (Note:  $X$  and  $y$  are feature matrix and label vector respectively.)

```
from sklearn.naive_bayes import CategoricalNB
model = CategoricalNB()
model.fit(X,y)
```

We find that the performance of the model is unsatisfactory. Which of the following models may be useful to use instead of CategoricalNB?

**Options :**

A. ✗ NumericalNB

B. ✗ MultinomialNB

C. ✗ BernoulliNB

D. ✓ GaussianNB

**Question Number : 80 Question Type : MCQ**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Consider the following two models:

Model A:

```
import pandas as pd
from sklearn.datasets import load_wine
from sklearn.linear_model import LogisticRegression
data = load_wine()
X = pd.DataFrame(data.data, columns = data.feature_names)
y = pd.DataFrame(data.target)
model = LogisticRegression(penalty = 'none', solver = 'saga', random_state = 100)
model.fit(X,y)
model.score(X,y)
```

Model B:

```
import pandas as pd
from sklearn.datasets import load_wine
from sklearn.linear_model import LogisticRegression
data = load_wine()
X = pd.DataFrame(data.data, columns = data.feature_names)
y = pd.DataFrame(data.target)
model = LogisticRegression(penalty = 'l2', solver = 'saga', random_state = 100)
model.fit(X,y)
model.score(X,y)
```

In general, which of the models will have a high bias?

**Options :**

- A. ✖ Model A
- B. ✔ Model B
- C. ✖ Both the models will have same bias
- D. ✖ Can't say

**Question Number : 81 Question Type : MCQ**

**Correct Marks : 2**

Question Label : Multiple Choice Question

The below figure (Figure 1) shows AUC-ROC curves for three logistic regression models. Different colors show curves for different hyper parameters values.

Which of the AUC-ROC curves will give best result?

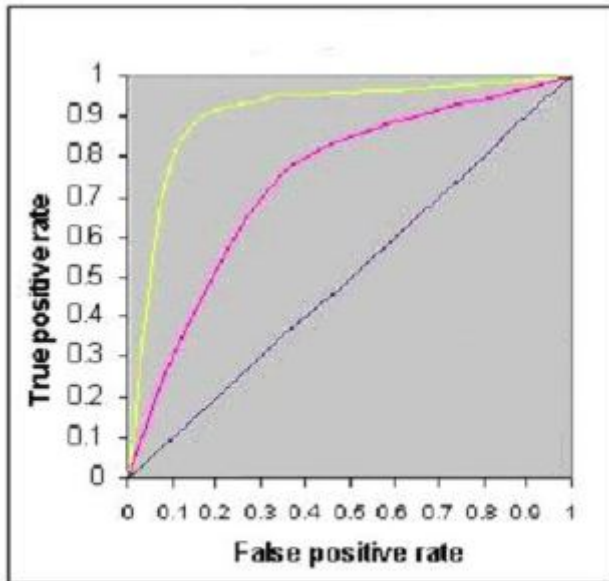


Figure 1

**Options :**

- A. ✓ Yellow
- B. ✗ Pink
- C. ✗ Blue
- D. ✗ All are same

**Question Number : 82 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 :**

- A. ✖ 'log', Logistic Regressor
- B. ✖ 'log', Logistic Classifier
- C. ✔ 'hinge', SVM Classifier
- D. ✖ 'hinge', Perceptron

**Question Number : 83 Question Type : MCQ**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Which of the following will be the correct output of the code snippet given below?

```
from sklearn.neighbors import KNeighborsClassifier
X = [[10], [15], [20], [25], [30]]
y = [0, 0, 1, 1, 2]
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X, y)
print(neigh.predict([[18]]))
```

**Options :**

- A. ✖ 0
- B. ✔ 1
- C. ✖ 2
- D. ✖ None

**Question Number : 84 Question Type : MCQ**

**Correct Marks : 2**

Question Label : Multiple Choice Question

What would be the best value for k to be used in KNN algorithm based on the graph given below?

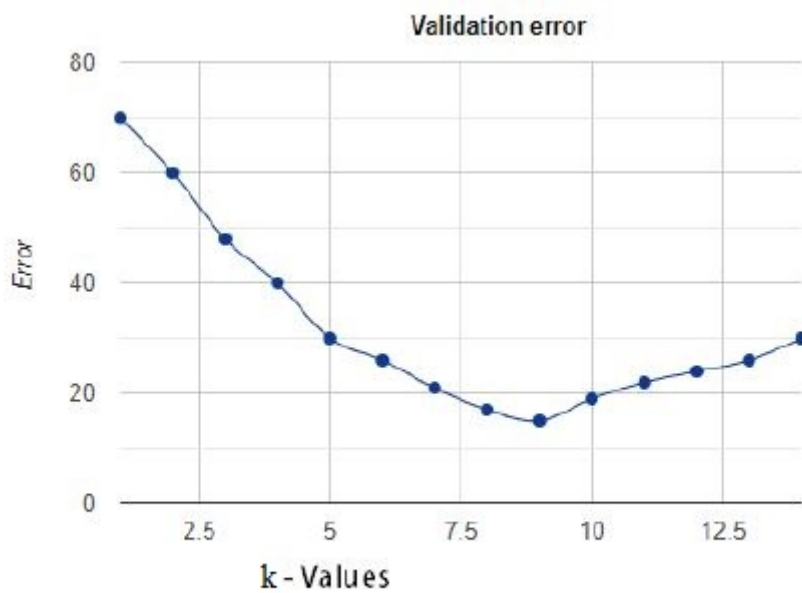


Figure 2

**Options :**

- A. ✗ 4
- B. ✗ 5
- C. ✓ 9
- D. ✗ 12

**Question Number : 85 Question Type : MCQ**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Which of the following numerical value should be at the place of "?" symbol in the output? (Assume necessary imports)

```
import pandas as pd
text= ['An examination is an official test','It shows knowledge in particular subject ']
countvectorizer = CountVectorizer()
est = countvectorizer.fit_transform(text)
Words = countvectorizer.get_feature_names()
df = pd.DataFrame(data = est.toarray(),index = ['Sentence1','Sentence2'],columns = Words)
df.head()
```

**Output:**

		an	examination	in	is	it	knowledge	official	particular	shows	subject	test
Sentence1	?			1	0	1	0	0	1	0	0	1
Sentence2	0			0	1	0	1	1	0	1	1	0

Figure 3

**Options :**

- A. ✖ 0
- B. ✖ 1
- C. ✔ 2
- D. ✖ None

**Question Number : 86 Question Type : MCQ**

**Correct Marks : 2**

Question Label : Multiple Choice Question

What is the output of the following block of code?

```
import numpy as np
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
X = np.array([[1, 0], [0, 1]])
y = np.array([-1, 1])
from sklearn.svm import SVC
clf = make_pipeline(StandardScaler(), SVC(kernel='linear'))
clf.fit(X, y)
print(clf.predict([[-2,2]]))
```

**Options :**

- A. ✖ array[1]
- B. ✖ array[3]
- C. ✖ array[2]



D. ✓ array[-1]

**Question Number : 87 Question Type : MCQ**

**Correct Marks : 3**

Question Label : Multiple Choice Question

What is the output of following code snippet (assume necessary imports)?

```
X,y= fetch_openml('mnist_784',version=1,return_X_y=True)
X = X.to_numpy()
y = y.to_numpy()

indices1 = np.where(y == 8)[0]
indices2 = np.where(y == 0)[0]

y_train = y[np.concatenate((indices1, indices2), axis = 0)]
X_train = X[np.concatenate((indices1, indices2), axis = 0)]
```

**Options :**

- A. ✓ it prepares data for binary classification of 0 and 8
- B. ✗ it prepares data for binary classification of 6 and 9
- C. ✗ it prepares data for multi class classification of 1 to 7 and 9
- D. ✗ none of these

**Question Number : 88 Question Type : MCQ**

**Correct Marks : 3**

Question Label : Multiple Choice Question

A perceptron model is trained on binary output data  $X_{n \times m}$  and  $y_{n \times 1}$ .

```
clf = Perceptron()
clf.fit(X,y)
```

Which of the following will plot the PR curve correctly?

**Options :**



A. ✓ 

```
scores = clf.decision_function(X)
p,r, _ = precision_recall_curve(y, scores)
plt.plot(p[:-1], r[:-1])
```

B. ✗ 

```
s = clf.decision_function(X)
p,r, _ = PrecisionRecallCurve(y, s)
plt.plot(p[:-1], r[:-1])
```

C. ✗ 

```
s = clf.decision_score(X)
p,r, _ = precision_recall_curve(s, y)
plt.plot(p[:-1], r[:-1])
```

D. ✗ 

```
s = clf.decision_score(X)
p,r, _ = precision_recall_curve(y, s)
plt.plot(p[:-1], r[:-1])
```

**Question Number : 89 Question Type : MCQ**

**Correct Marks : 3**

Question Label : Multiple Choice Question

We have run the following code on a highly imbalanced dataset and we observe that it has performed poorly on parameters like accuracy, precision and recall.

```

import pandas as pd
from sklearn.metrics import classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline

df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases
                /wine-quality/winequality-red.csv', sep = ';')

X = df.iloc[:,0:11].copy()
Y = df.iloc[:,11:].copy()
X_train,X_test,y_train,y_test = train_test_split(X,Y, test_size = 0.2)
lr = LogisticRegression(random_state = 1)
lr.fit(X_train, y_train)
lr_cp = classification_report(y_test, lr.predict(X_test), output_dict = True)
lr_cp = pd.DataFrame(lr_cp).T
lr_cp

```

Which among the following techniques is most likely to improve the accuracy of the model?

**Options :**

- A. ✖ By using a different solver.
- B. ✖ By passing an appropriate value of l1\_ratio.
- C. ✔ By passing an appropriate value for class\_weight.
- D. ✖ By increasing the number of iterations.

**Question Number : 90 Question Type : MCQ**

**Correct Marks : 3**

Question Label : Multiple Choice Question

We have run the following code:

```
import numpy as np
from sklearn.linear_model import LogisticRegression
x1 = 7*np.random.randn(50)
x2 = 8*np.random.randn(50)
x3 = 9*np.random.randn(50)
y = (10 + x1 + x2 + 0.5*np.random.randn()) > 0
X = np.column_stack([x1, x2,x3])
lr = LogisticRegression(solver = 'liblinear', tol = 1e-4, penalty = 'l2',
                        fit_intercept = False, dual = True, random_state = 1)

lr.fit(X, y)
lr.score(X,y)
```

Knowing the details of the data generation, by changing which of the following hyperparameters, we are likely to have the highest improvement in score?

**Options :**

- A. ✖ By changing solver to 'saga' and making penalty as 'elasticnet'.
- B. ✖ By changing dual to False and increasing tol to 1e-3.
- C. ✔ By changing fit intercept to True.
- D. ✖ By changing penalty alone to 'l1'.

**Question Number : 91 Question Type : MCQ**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Rahul is working on a large dataset where he needs to Convert a collection of text documents to a matrix. Which of the following vectorizer would you suggest him to use?

**Options :**

- A. ✖ CountVectorizer
- B. ✖ TfidfVectorizer
- C. ✔ HashingVectorizer
- D. ✖ Both CountVectorizer and HashingVectorizer

**Question Number : 92 Question Type : MSQ**

**Correct Marks : 2**

Question Label : Multiple Select Question

Consider following code and choose correct options (assume necessary imports).

```
ridge_classifier = RidgeClassifier(alpha=0.0001,  
                                  fit_intercept=True)
```

**Options :**

- A. ✓ It fits a model with an intercept.
- B. ✓ The L2 regularization rate is 0.0001.
- C. ✗ It fits a logistic regression model.
- D. ✗ It is suitable for fitting a regression model.

**Question Number : 93 Question Type : MSQ**

**Correct Marks : 2**

Question Label : Multiple Select Question

Vishal is working on a classification model. He is having the dataset of 120 GB size. But the ram of his machine is 4GB only. Which of the following algorithms do you think can help him? Select all that apply.

**Options :**

- A. ✗ SGDRegressor
- B. ✓ Perceptron
- C. ✗ LogisticRegression
- D. ✓ SGDClassifier
- E. ✓ MultinomialNB

**Question Number : 94 Question Type : MSQ**

**Correct Marks : 3**

Question Label : Multiple Select Question

Which of the following code will correctly train a large scale regression model using partial\_fit, If

shape of X\_train,Y\_train are given as (9000,90,10) and (9000,90) respectively?

**Options :**

A. ✖ 

```
from sklearn.linear_model import SGDRegressor
regressor = SGDRegressor(random_state=10)
for i in range(X_train.shape[[0]]):
    X_batch, Y_batch = X_train[i], Y_train[i]
    regressor.partial_fit(X_batch, Y_batch)
```

B. ✔ 

```
from sklearn.linear_model import SGDRegressor
regressor = SGDRegressor(random_state=10)
for i in range(Y_train.shape[0]):
    X_batch, Y_batch = X_train[i], Y_train[i]
    regressor.partial_fit(X_batch, Y_batch)
```

C. ✖ 

```
from sklearn.linear_model import SGDRegressor
regressor = SGDRegressor(random_state=10)
for i in range(Y_train.shape[[0]]):
    X_batch, Y_batch = X_train[i], Y_train[i]
    regressor.partial_fit(X_batch, Y_batch)
```

D. ✔ 

```
from sklearn.linear_model import SGDRegressor
regressor = SGDRegressor(random_state=10)
for i in range(X_train.shape[0]):
    X_batch, Y_batch = X_train[i], Y_train[i]
    regressor.partial_fit(X_batch, Y_batch)
```

**Question Number : 95 Question Type : MSQ**

**Correct Marks : 3**

Question Label : Multiple Select Question

The one-vs-one approach in multi-class classification is implemented by-

**Options :**

A. ✖ LinearSVC

B. ✔ SVC

C. ✔ NuSVC

## D. ✖ MultiSVC

Question Number : 96 Question Type : SA

Correct Marks : 2

Question Label : Short Answer Question

Abhishek wrote the following code to train the Figure 4:

```
from sklearn.svm import SVC
SVC_classifier = SVC()
clf = SVC_classifier.fit(X_train, y_train)
print(clf.support_vectors_)
```

what is the output he will get?

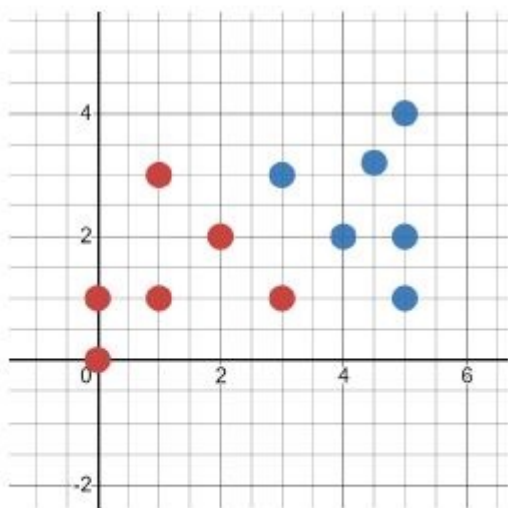


Figure 4

**NOTE:** Enter your answer to the nearest integer.

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Equal

**Text Areas :** PlainText

**Possible Answers :**

6



**Question Number : 97 Question Type : SA**

**Correct Marks : 2**

Question Label : Short Answer Question

What is the output of the following code?

```
from sklearn import svm
X = [[0], [1], [2], [3]]
Y = [0, 1, 2, 3]
clf = svm.SVC(decision_function_shape='ovo')
clf.fit(X, Y)
dec = clf.decision_function([[1]])
dec.shape[1]
```

**NOTE:** Enter your answer to the nearest integer.

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Equal

**Text Areas :** PlainText

**Possible Answers :**

6

**Question Number : 98 Question Type : SA**

**Correct Marks : 3**

Question Label : Short Answer Question

What will be the output of the following code snippet?

```
from sklearn.metrics import DistanceMetric
dist = DistanceMetric.get_metric('manhattan')
X = [[3, 4,],
      [5, 6,]]
print(dist.pairwise(X)[0][1])
```



**NOTE:** Enter your answer to the nearest integer.

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Equal

**Text Areas :** PlainText

**Possible Answers :**

4

## MLT

**Number of Questions :** 15

**Section Marks :** 50

**Question Number : 99 Question Type : MCQ**

**Correct Marks : 0**

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL: MACHINE LEARNING TECHNIQUES"

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 :**

A. ✓ Yes

B. ✗ No