Preamble: Load the dataset and examine it.

Q1. [marks : 0] Which dataset are you using for this exam?

[MCQ]

Options:

A) v1

B) v2

C) v3

D) v4

E) v5

Answer: v1: A, v2:B, v3: C, v4:D, v5:E

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
rs=32 #assigned a random state
```

Q2: [marks: 2] Load the dataset. What is its shape?

[MCQ]

Options:

A) (1000, 18)

B) (1000, 20)

C) (900, 19)

D) (1000, 19)

E) (900, 20)

Answer: D) for all versions

```
from google.colab import files
files.upload()
```

```
#data = pd.read_csv('v1.csv')
data = pd.read_csv('v2.csv')
#data = pd.read_csv('v3.csv')
#data = pd.read_csv('v4.csv')
# data = pd.read_csv('v5.csv')
```

```
data.head()
```

	Country	Status	Life_expectancy	Adult_Mortality	infant_deaths	Alcohol	perc
0	Ecuador	Developing	75.1	137.0	7	3.87	
1	Suriname	Developing	70.0	196.0	0	5.13	
2	Togo	Developing	59.7	285.0	13	0.01	
3	United States of America	Developed	79.1	14.0	23	8.82	
4	Philippines	Developing	67.2	217.0	67	4.44	

```
data.shape
```

```
(1000, 19)
```

```
data.describe()
```

Q3: [marks: 2] Are there any missing values in the column 'Life_expectancy'? **If there are, remove the corresponding rows from the data. Note that this modified data will be used for the subsequent questions.**

[MCQ]

Options:

A) 5

B) 2

C) 4

D) 3

E) 0

Answer: D (v1), B (v2), C (v3), A (v4), C (v5)

```
data = data[data.Life_expectancy.notna()]
```

```
data.describe()
```

	Life_expectancy	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditu
count	998.000000	998.000000	998.000000	933.000000	998.0000
mean	69.187976	162.061122	31.581162	4.647503	792.3516
std	9.697537	121.590402	130.487493	4.019032	2148.2707
min	36.300000	1.000000	0.000000	0.010000	0.0000
25%	62.925000	72.000000	0.000000	1.000000	4.9054
50%	72.250000	138.000000	3.000000	3.890000	59.5931
75%	75.975000	228.000000	21.750000	7.840000	430.4538
max	89.000000	715.000000	1800.000000	16.990000	18961.3486

Q4: [marks: 1] How many categorical features are there in the dataset?

[MCQ]

Options:

A) 1

B) 2

C) 3

D) 4

E) 5

Answer: B) for all versions

```
data.head()
```

	Country	Status	Life_expectancy	Adult_Mortality	infant_deaths	Alcohol	perc
0	Ecuador	Developing	75.1	137.0	7	3.87	
1	Suriname	Developing	70.0	196.0	0	5.13	
2	Togo	Developing	59.7	285.0	13	0.01	
3	United States of America	Developed	79.1	14.0	23	8.82	
4	Philippines	Developing	67.2	217.0	67	4.44	

Q5: [marks: 1] What is the average BMI in the dataset?

[MCQ]

A) 37.125355

B) 37.632287

C) 38.150305

D) 38.063931

E) 39.292510

Answer: C (v1), A(v2), E (v3), D (v4), B (v5)

```
data.describe()
```

	Life_expectancy	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditu
count	998.000000	998.000000	998.000000	933.000000	998.0000
mean	69.187976	162.061122	31.581162	4.647503	792.3516
std	9.697537	121.590402	130.487493	4.019032	2148.2707
min	36.300000	1.000000	0.000000	0.010000	0.0000
25%	62.925000	72.000000	0.000000	1.000000	4.9054
50%	72.250000	138.000000	3.000000	3.890000	59.5931
75%	75.975000	228.000000	21.750000	7.840000	430.4538
max	89.000000	715.000000	1800.000000	16.990000	18961.3486

Q6 [marks: 2] How many missing values are there in the columns 'Hepatitis_B' and 'Population'?

[MCQ] Options:

A) 179, 225

B) 192, 218

C) 195, 229

D) 196, 209

E) 172, 209

Answer: D (v1), C (v2), A (v3), E (v4), B (v5)

```
data.isnull().sum()
```

```
Country                          0
Status                           0
Life_expectancy                  0
Adult_Mortality                  0
infant_deaths                    0
Alcohol                         65
percentage_expenditure           0
Hepatitis_B                    195
Measles                          0
BMI                             12
under-five_deaths                0
```

```
Polio                               6
Total_expenditure                  75
Diphtheria                          6
HIV_AIDS                            0
GDP                               149
Population                        229
Income_composition_of_resources    53
Schooling                          53
dtype: int64
```

Q7: [marks: 1] How many unique countries are there in the dataset?

[MCQ] Options:

A) 187

B) 180

C) 189

D) 183

E) None of these

Answer: D (for all versions)

```
len(data['Country'].unique())
```

```
183
```

Q8: [marks: 3] The column 'Life_expectancy' is to be used as the target column. Split the data into the feature matrix (X) and target column (y), where 'Life_expectancy' goes to y and rest of the columns go to X. **What is the average 'Life_expectancy' in the dataset?**

[MCQ] Options:

A) 69.183333

B) 69.556928

C) 68.842929

D) 69.295578

E) 69.187976

Answer: C (v1), E (v2), B (v3), D (v4), A (v5)

```
data.shape
```

```
(998, 19)
```

```
y = data.loc[:, ['Life_expectancy']]
X = data.drop(['Life_expectancy'], axis = 1)
```

```
data.shape
```

```
(998, 19)
```

```
y.shape
```

```
(998, 1)
```

```
X.shape
```

```
(998, 18)
```

```
y
```

	Life_expectancy
0	75.1
1	70.0
2	59.7
3	79.1
4	67.2
...	...
995	78.7
996	73.2
997	75.8
998	78.3
999	78.3

998 rows × 1 columns

```
y.mean()
```

```
Life_expectancy    69.187976
dtype: float64
```

Q9: [marks: 2] Split X and y into X_train, X_test, y_train and y_test where 20% of the data goes to test set. Keep the random_state to be 32. What is the average value of GDP in training and test data (rounded to 2 decimal places)?

[MCQ]

Options:

A) 7449.13, 6571.79

B) 8004.37, 5871.91

C) 7928.45, 7845.42

D) 7677.76, 8057.68

E) 6347.62, 8154.31

Answer: B (v1), D (v2), C (v3), A (v4), E (v5)

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size = 0.2,
                                                    random_state = rs)
```

```
type(X_train)
```

```
    pandas.core.frame.DataFrame
```

```
X_train.GDP.mean()
```

```
    7677.768313354994
```

```
X_test.GDP.mean()
```

```
    8057.684001866457
```

Q10: [marks: 2] Plot the distribution of different numerical features in the training data. Which of the following features has close to normal distribution?

[MCQ]

Options:

A) Percentage_expenditure

B) Measles

C) Under-five_deaths

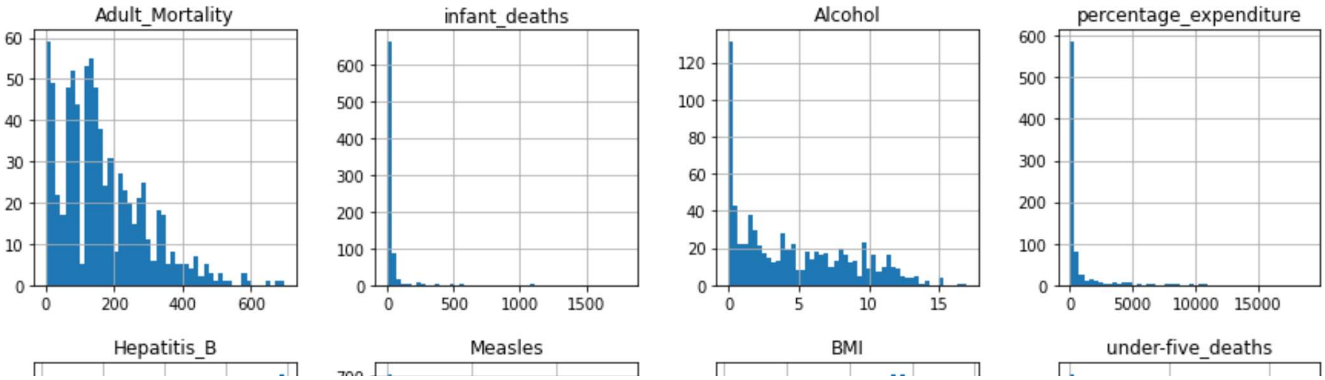
D) HIV_AIDS

E) Schooling

Answer: Schooling (for all versions)

```
X_train.hist(bins=50,figsize=(15,15))
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95ea7450>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95d11a10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95ccbfd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95c8c650>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95cc3c50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95c86290>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95c3c910>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95bf3e50>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95bf3e90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95bb65d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95b34550>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95af5d50>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95aaaed0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95a6c550>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c95a21b50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f5c959e5190>]],
      dtype=object)
```



Week 2

Q11 (4 marks)

For the feature `Diphtheria` in the feature matrix training set `X_train`, draw a violin plot and find which of the following ranges hold most of the values?

[MCQ]

(a) 20-40

(b) 40-60

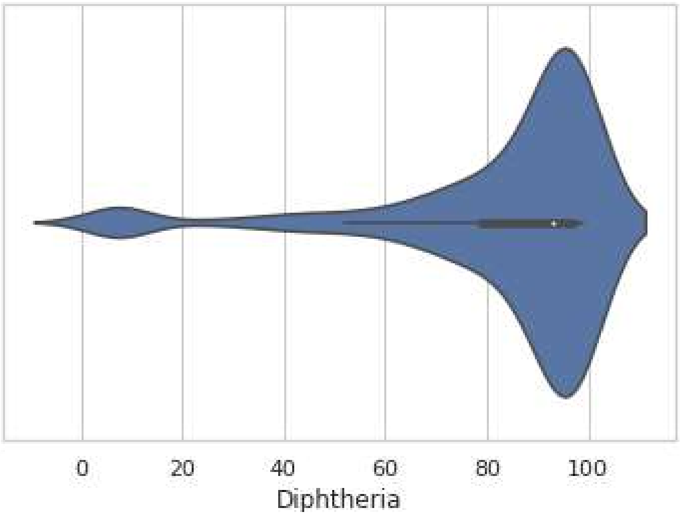
(c) 60-80

(d) 80-100

Ans: Option (d) : V1, V2, V3, V4, V5

```
import seaborn as sns
sns.set_theme(style="whitegrid")
```

```
#tips = sns.load_dataset("tips")
ax = sns.violinplot(x=X_train["Diphtheria"])
```



Q12 (2 Marks)

By plotting a box plot for the numerical features of the feature matrix training set `X_train`, find out which of the following features have no outliers? [MCQ]

(a) `Alcohol`

(b) `under-five_deaths`

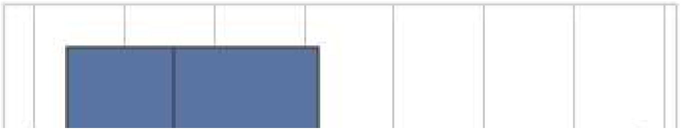
(c) `Measles`

(d) `GDP`

(e) None of these

Answer : Option (a) : V1, V2, V3 Option (e) : V4, V5

```
import seaborn as sns
sns.set_theme(style="whitegrid")
#tips = sns.load_dataset("tips")
#ax = sns.boxplot(x=X["Alcohol"])
ax = sns.boxplot(x=X_train["Alcohol"])
```



Q13 (2 Marks)

Generate a new matrix consisting of all polynomial combinations of the features with degree 2 (`For example, if an input sample is two dimensional and of the form [a,b] , the degree-2 polynomial features are` $[1, a, b, a^2, ab, b^2]$) from the training set of feature matrix `X_train`. Fit and transform the training set of the feature matrix columns `[2:4]` and save it with the name `polydata`, after applying the polynomial transformation. Note that the `polydata` will not be utilized in corresponding questions. Choose the shape of `polydata` from the following options.

[MCQ]

(a) (799, 6))

(b) (797,6)

(c) (798,6)

(d) (796,6)

Answer : option B : V1, option C : V2, option d : V3, V5, V4

```
from sklearn.preprocessing import PolynomialFeatures
print('Number of features before transformation = ', X_train[X_train.columns[2:4]].shape)

poly = PolynomialFeatures(degree=2)
polydata = poly.fit_transform(X_train[X_train.columns[2:4]])
print('Number of features after transformation = ', polydata.shape)
```

```
    Number of features before transformation =  (798, 2)
    Number of features after transformation =  (798, 6)
```

Q14: (Marks : 6)

Prepare a pipeline `numeric_transformer` containing `SimpleImputer (strategy="mean")` and `StandardScaler()` (in this sequence). Preprocess the `numeric_features` of the given data ('Adult_Mortality', 'infant_deaths', 'Alcohol', 'percentage_expenditure', 'Hepatitis_B', 'Measles', 'BMI', 'under-five_deaths', 'Polio', 'Total_expenditure', 'Diphtheria', 'HIV_AIDS', 'GDP', 'Population', 'Income_composition_of_resources', 'Schooling') using this pipeline. Apply `categorical_transformer = OneHotEncoder()` on the `categorical_features` ('Country', 'Status') and other features will pass unchanged.

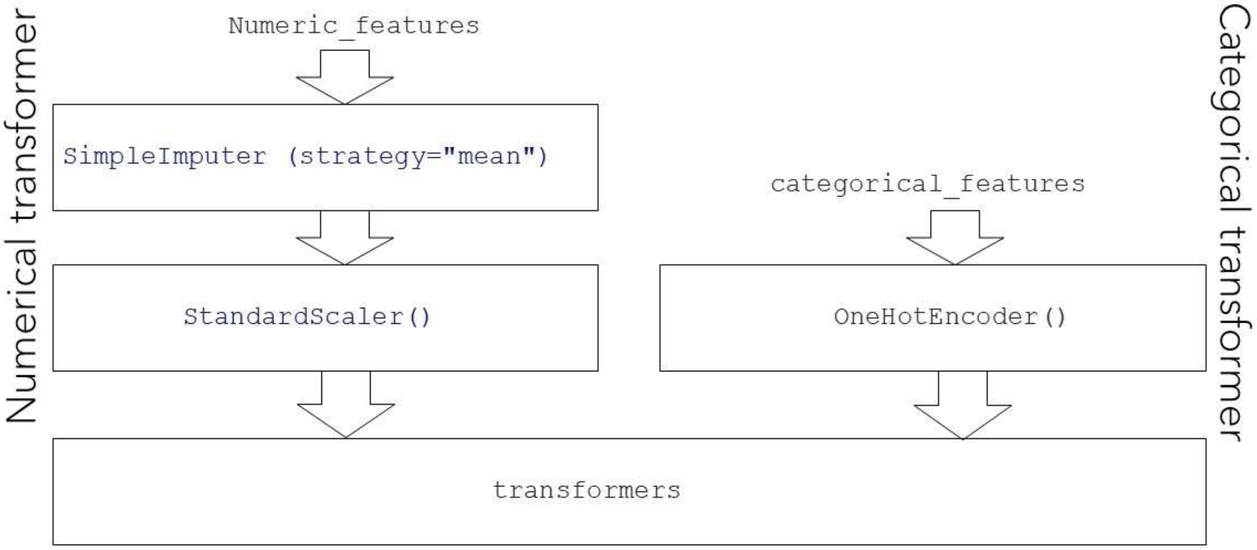
IMPORTANT NOTE:

1. The data obtained by transforming via this pipeline will be used for the rest of the questions.

2. Use the pipeline to preprocess training data and then apply on test data

What is the length of the `numeric_transformer` pipeline?

[NAT]



Answer : 2

```
numeric_features = ['Adult_Mortality',
                    'infant_deaths',
                    'Alcohol',
                    'percentage_expenditure',
                    'Hepatitis_B', 'Measles',
                    'BMI',  'under-five_deaths',
                    'Polio',    'Total_expenditure',
                    'Diphtheria',   'HIV_AIDS',
                    'GDP',  'Population',
                    'Income_composition_of_resources',
                    'Schooling']
categorical_features = ['Country',  'Status']
# hypertension and heart_disease are binary only.


from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline

numeric_transformer = Pipeline(
    steps=[("imputer", SimpleImputer(missing_values = np.nan, strategy="mean")),
           ("scaler", StandardScaler())]
)
categorical_transformer = OneHotEncoder(handle_unknown='ignore')


from sklearn.compose import ColumnTransformer

preprocessor = ColumnTransformer(
    transformers=[
        ("num", numeric_transformer, numeric_features),
        ("cat", categorical_transformer, categorical_features),
    ], remainder='passthrough'
)


X_train = preprocessor.fit_transform(X_train)
X_test = preprocessor.transform(X_test)
```

```
print(len(numeric_transformer.steps))
```

```
2
```

```
X_train.shape
```

```
(798, 200)
```

```
type(X_train)
```

```
scipy.sparse.csr.csr_matrix
```

Q15 [Marks : 2] Calculate the shape of the training set of feature matrix `X_train` of the dataset.

[MCQ]

(a) (796, 201)

(b) (797, 201)

(c) (798, 200)

(d) (796, 200)

(e) (800,204)

Option (B): V1, Option (c) : V2, Option (a) : V4, option (d) : V5, V3

```
print(X_train.shape)
print(X.shape)
print(y.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
     (798, 200)
     (998, 18)
     (998, 1)
     (200, 200)
     (798, 1)
     (200, 1)
```

```
type(X_train)
```

```
     scipy.sparse.csr.csr_matrix
```

```
type(y_train)
```

```
     pandas.core.frame.DataFrame
```

```
X_train = X_train.toarray()
```

```
 X_train.shape
```

```
     (798, 200)
```

```
y_train.shape
```

```
     (798, 1)
```

```
y_train.isnull().sum()
```

```
     Life_expectancy    0
     dtype: int64
```

```
X_train.shape
```

```
     (798, 200)
```

Q16 (4 marks):

Apply RFE and select 15 features from the training dataset and calculate the rank of each feature. Which of the following statements are true? Count first feature as number 1, second feature as number 2 and so on.

Note: The subsequent questions should not use the dataset reduced by this question and use the previous data.

[MCQ]

(a) The second feature from the beginning is selected

(b) The fifth feature is selected.

(c) The eight-th feature is selected.

(d) Only 14 features can be selected.

(e) None of the features 2, 5, 8 is selected

V1, V2, V3, V4, V5: Option (a), (c)

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
estimator = LinearRegression()
selector = RFE(estimator, n_features_to_select = 15, step=1)
selector = selector.fit(X_train,y_train)
# support_ attribute is a boolean array
# marking which features are selected
print(selector.support_)
# rank of each feature
# if it's value is '1', then it is selected
# features with rank 2 and onwards are ranked least.
print(f'Rank of each feature is : {selector.ranking_}')
```

```
[False  True False False False False False  True False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False  True  True
 False False False False False False False False False False False  True
 False False False False False False False False False False False False
 False  True  True False False False False False  True False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False  True
 False False False False False  True False False False False False False
 False False False False False False False False False False False False
 False  True False False False False False False False False False False
 False  True False False False False False False False False False False
 False False  True False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False  True False  True]
Rank of each feature is : [175   1 167 182 181 180 185   1 183 184 179 135 178 186 168 1
  35  96  14  23  36  73  82  20  18   8 171  19 174  86 169 109 156 144
  11  98 176  10  58 120 101  47 155 105   1   1  94   3 165  92 129 127
   2  64   4  88  66   1 115 110  85 145  52  17 116  89  95 114  30 160
 157   1   1 140 113  27  87 130   1  41 124 107  99 152 106  38  61  68
 121 133  44  91  84  71  24  72  90 136 122 142  28 162 128  62  29   1
 111 172  57  79 131   1  32  16 119  67 147  45  33 177 143  22 118  97
 159 148 149  83  70  39 164   1  74  26  93   6 141  56  31 158  63  80
   5   1 173  59 134 126  15  12  46 150  53 146  37  55   1  76  60  81
 170 100 138 104  75  51 151 123 154  69  78  43 161  49  34 132 102  54
 117  25  42 139 125 166  13  77 108  65   9 163  40  48  21 153 103   1
  50   1]
```

Week-3

Common data for Q17-Q19.

Do not change any default value of the parameters for the model.

Q17 [5 marks]

Fit a `LinearRegression` model that uses the normal equation to learn the weights on the train data (`X_train` and `y_train`). Enter the value of `score` obtained using training data upto four decimal places.

Answer: NAT
v1: 0.9706 Range [0.96, 0.98]
v2: 0.9507 Range [0.94, 0.96]
v3: 0.9606 Range [0.95, 0.97]
v4: 0.9694 Range [0.96, 0.98]
v5: 0.9677 Range [0.95, 0.98]

Solution

Q18 [5 marks]

Using a Linear regression model, compute the cross-validation scores for 15 splits on training data (`X_train` and `y_train`) using `cross_val_score`. Enter the maximum value obtained upto four decimal places. By default `cross_val_score` uses `LinearRegression`'s scoring metric, which is R2.

Answer: NAT

v1: 0.9811 Range [0.97, 0.99]

v2: 0.9512 Range [0.94, 0.96]

v3: 0.9749 Range [0.96, 0.98]

v4: 0.9632 Range [0.95, 0.97]

v5: 0.9685 Range [0.96, 0.98]

Solution

Q19 [10 marks]

Fit a `Stochastic Gradient Descent` regressor model on the training data (`X_train` and `y_train`).
Set the following parameters:
(i) penalty = l1
(ii) alpha = 0.001
(iii) learning_rate = 'constant'
(iv) initial learning rate = 0.001
(v) random state = 42
Other parameters are initialized with default values.
Compute the `mean_squared_error` for training data. Using the trained model, make predictions on test data and then compute the `mean_squared_error` for the test data also. Find the absolute difference between the training and testing errors computed and enter the value upto two decimal places, i.e. $|MSE_{train} - MSE_{test}|$.

Answer: NAT

v1: 6.26 Range [6.2, 6.3]

v2: 3.43 Range [3.4, 3.5]

v3: 1.96 Range [1.9, 2]

v4: 2.89 Range [2.85, 2.95]

v5: 1.76 Range [1.7, 1.8]

Solution

Q20 [5 marks]

Train a `Stochastic Gradient Descent` regressor model on the training data (`X_train` and `y_train`) with different values for the parameters as follows:
(i) penalty = l2
(ii) alpha = 0.01

(iii) learning_rate = 'adaptive'

(iv) initial learning rate = 0.01

(v) random state = 42

(vi) loss = 'huber'

Other parameters are initialized with default values.

Using the trained model, make predictions on test data and then compute the `r2_score` for the test data. Enter the value upto four decimal places.

Answer: NAT

v1: 0.7602 Range [0.758, 0.765]

v2: 0.7789 Range [0.775, 0.781]

v3: 0.8043 Range [0.795, 0.81]

v4: 0.7727 Range [0.769, 0.779]

v5: 0.8066 Range [0.795, 0.81]

Solution

Q21 [5 marks]

Apply cross validation strategy on SGD regression model with parameters same as that of previous question using `ShuffleSplit` with 10 number of splits and 0.2 test size on 'train data'. Use `random_state=42` for `ShuffleSplit`. Enter the standard deviation value of `cross_val_score` obtained upto four decimal places.

Answer: NAT

v1: 0.0284 Range [0.02, 0.03]

v2: 0.0140 Range [0.0135, 0.0145]

v3: 0.0159 Range [0.0155, 0.0165]

v4: 0.0346 Range [0.03, 0.04]

v5: 0.0169 Range [0.016, 0.0175]

Solution

Week-4

Q22: [marks: 2] Take Lasso estimator with regularization rate 0.05 to train the model using Training data. What would be the value of Mean Squared error for test data?(Set random state =42)

Ans:

for v1: 19.95 Range 17-22

for v2: 18.62 Range 16-21

for v3: 13.41 Range 11-16

for v4: 17.67 Range 15-20

for v5: 14.37 Range 12-17

Q23 [4 marks] Create a baseline model using Ridge estimator with fixed learning rate 0.5. What is the R2 score you got on training data(Set random state =32)? [NAT]

Ans:

for v1: 0.965 Range 0.955-0.975

for v2: 0.9522 Range 0.945-0.965

for v3: 0.9541 Range 0.94-0.96

for v4: 0.96 Range 0.95-0.975

for v5: 0.9605 Range 0.95-0.975

Q24 [4 marks] Using above baseline Ridge model with fixed regularization rate 0.5, Predict the R2 score for the test data? [NAT]

Ans:

for v1: 0.9151 Range0.90-0.92

for v2: 0.922 Range 0.91-0.933

for v3: 0.9161 Range0.90-0.92

for v4: 0.9362 Range 0.92-0.95

for v5: 0.9409 Range 0.92-0.95

Common Instructions for Q25, Q26

i) Use Ridgecv as an estimator to train the model.

i) Use following list for alpha values. alpha_list = [1e-4,1e-3, 1e-2, 1e-1, 1].

ii) Keep r2_score as scoring parameter.

iii) Keep Cross validation iterator "RepeatedKFold" for cv and keep splitting iterations parameter n_split=5,n_repeats=5 and random_state=32.

iv) Use Training data(X_train, y_train) for model training.

Q25 [6marks] Which of the following alpha value gives the best R2 score for training data?

[MCQ]

Options:

1. 1e-4
2. 1e-3
3. 1e-2
4. 1e-1
5. 1

Ans:

Option 4 for v1,v2,v3,v5 version

Option 2 for v4 version

Q26 [4 marks] What is the best r2_score you got with best alpha value for training data?(Choose closest value) [NAT]

Ans:

- for v1: 0.935 range: 0.925-0.945
- for v2: 0.914 range: 0.90-0.92
- for v3: 0.910 range: 0.90-0.92
- for v4: 0.931 range: 0.925-0.945
- for v5: 0.914 range: 0.90-0.92

Common Instructions for Q27 - Q28:

Create a pipeline Using PolynomialFeatures as transformer and Lasso as estimator. Use gridsearch with above pipeline and following hyperparameter values.

i) Keep polynomial degree as : (1, 2)

ii) alpha value : np.logspace(-4, 0, num=10)

iii) scoring : neg_mean_absolute_error

iv) Use Training data(X_train, y_train) to fit the model.

Q27 [6 marks] Enter the regularization rate which gives the lowest mean square error value for training data ? [NAT]

Ans:

- for v1: 0.00027 range: 0.0002- 0.0003
- for v2: 0.0001 range: 0.0- 0.0002
- for v3: 0.00027 range: 0.0002- 0.0003
- for v4: 0.00027 range: 0.0002- 0.0003
- for v5: 0.000774 range: 0.0006 -0.0008

Q28 [4 marks] What is the best score you got for training data using best alpha value ? [NAT]

Ans:

- for v1: -1.09 range [-1.15 , -1.05]
- for v2: -1.28 range [-1.35, -1.20]
- for v3: -1.2 range [-1.25, -1.15]
- for v4: -1.11 range [-1.15,-1.05]
- for v5: -1.13 range [-1.20,-1.10]

✓ 3m 5s completed at 11:54 AM