VALIDATION MODEL OF CLASSIFICATION ANALYSIS

(CLASSIFICATION OF IRIS FLOWER)

Classify using KNN (k=5)

Import Library

```
1  import numpy as np
2  import pandas as pd
3
4  from sklearn.model_selection import ShuffleSplit
5  from sklearn.model_selection import KFold
6  from sklearn.model_selection import LeaveOneOut
7
8  from sklearn.neighbors import KNeighborsClassifier
9  from sklearn.metrics import precision_score
```

Import Dataset (Input & Output)

Import Dataset

```
# IMPORT DATASET
dataset = pd.read_csv('C:\Hanif Izzudi
dataset
```

	Data 1	Data 2	Data 3	Data 4	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Input Data

```
# Input Data
x = dataset.drop(["Class"], axis = 1)
x
```

	Data 1	Data 2	Data 3	Data 4
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

```
x = np.array(x)
x
```

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2]
```

Output Data

```
# Output Data
y = dataset["Class"]
y
```

```
0          Iris-setosa
1          Iris-setosa
2          Iris-setosa
3          Iris-setosa
4          Iris-setosa
              ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Class, Length: 150, dtype: object
```

```
y = np.array(y)
y
```

```
array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
```

Holdout method / “ShuffleSplit

```
1  rs = ShuffleSplit(n_splits=10, test_size=.25, random_state=0)
2  rs
```

```
ShuffleSplit(n_splits=10, random_state=0, test_size=0.25, train_size=None)
```

The Dataset is divided into 10 experiments, with splitting 25% for testing data and 75% for training data

```
==========> SHUFFLE SPLIT <==========
================================================================================
                                DATA SPLIT KE- 1
TRAIN: [ 61  92 112   2 141  43  10  60 116 144 119 108  69 135  56  80 123 133
 106 146  50 147  85  30 101  94  64  89  91 125  48  13 111  95  20  15
  52   3 149  98   6  68 109  96  12 102 120 104 128  46  11 110 124  41
 148   1 113 139  42   4 129  17  38   5  53 143 105   0  34  28  55  75
  35  23  74  31 118  57 131  65  32 138  14 122  19  29 130  49 136  99
  82  79 115 145  72  77  25  81 140 142  39  58  88  70  87  36  21   9
 103  67 117  47]
TEST: [114  62  33 107   7 100  40  86  76  71 134  51  73  54  63  37  78  90
  45  16 121  66  24   8 126  22  44  97  93  26 137  84  27 127 132  59
  18  83]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
0.9736842105263158
```

This is the example of output program in experiment 1. There are 112 training data and 38 testing data and this data is randomly selected.

The accuracy score of experiment 1 is 97.36%

The Program of KNN using Holdout method

```python
# Shuffle Split
print("==========> SHUFFLE SPLIT <==========")
i = 0
accuracy_model = []
for train_index, test_index in rs.split(x):
    print("="*80)
    i = i+1
    print("\t\t\t\t DATA SPLIT KE-", i)

    print("TRAIN:", train_index, "\nTEST:", test_index)

    x_train, x_test = x[train_index], x[test_index]
    y_train, y_test = y[train_index], y[test_index]
    #print("X_test: ", x_test)

    #KNN Classifier
    classifier = KNeighborsClassifier(n_neighbors=5)
    classifier.fit(x_train, y_train)

    #predict test data using KNN
    pred_out = classifier.predict(x_test)
    print("==> PREDICTION <==")
    #print(pred_out)
    print("-"*80)
    print("==> REAL OUTPUT <==")
    #print(y_test)
    print("-"*80)
    print("==> ACCURACY SCORE <==")
    accuracy = accuracy_score(y_test, pred_out)
    accuracy_model.append(accuracy)
    print(accuracy)
    print("="*80)
    accuracy_model.append(accuracy)

avg_accuracy = np.mean(accuracy_model)
print("==> AVERAGE ACCURACY = ", avg_accuracy)
```

```
==========================================================
==> AVERAGE ACCURACY =  0.9684210526315787
```

The average accuracy from 25% testing data each experiments is 96.84%

K-fold cross validation / “KFold"

```
1  kf = KFold(n_splits=10, shuffle=True)
2  kf
```

```
KFold(n_splits=10, random_state=None, shuffle=True)
```

The Dataset split with k=10 and shuffle it because the class of the dataset has sorted

```
==========> KFold <==========
================================================================================
                                DATA SPLIT KE- 1
TRAIN: [  0   1   2   3   4   5   7   8  10  11  12  13  14  15  16  17  18  19
  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
  38  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  57
  58  59  60  62  63  65  66  67  68  69  70  71  72  74  75  76  77  78
  79  80  81  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97
  98  99 100 102 103 104 105 106 107 108 109 110 111 112 113 114 115 117
 118 119 120 121 122 123 124 125 126 128 129 130 131 132 134 135 136 137
 138 139 141 142 143 144 146 147 149]
TEST: [  6   9  39  56  61  64  73  82 101 116 127 133 140 145 148]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
0.9333333333333333
```

This is the example of output program in experiment 1. There are 15 testing data, because the total data is 150 data, and the subsampling/”k” is 10, so (total data)/”k” is 15

The accuracy score of experiment 1 is 93.33%

The Program of KNN using K-fold cross validation

```python
# KFols
print("==========> KFold <==========")
i = 0
for train_index, test_index in kf.split(x):
    print("="*80)
    i = i+1
    print("\t\t\t\t DATA SPLIT KE-", i)

    print("TRAIN:", train_index, "\nTEST:", test_index)

    x_train, x_test = x[train_index], x[test_index]
    y_train, y_test = y[train_index], y[test_index]
    #print("X_test: ", x_test)

    #KNN Classifier
    classifier = KNeighborsClassifier(n_neighbors=5)
    classifier.fit(x_train, y_train)

    #predict test data using KNN
    pred_out = classifier.predict(x_test)
    print("==> PREDICTION <==")
    #print(pred_out)
    print("-"*80)
    print("==> REAL OUTPUT <==")
    #print(y_test)
    print("-"*80)
    print("==> ACCURACY SCORE <==")
    accuracy = accuracy_score(y_test, pred_out)
    accuracy_model.append(accuracy)
    print(accuracy)
    print("="*80)
    accuracy_model.append(accuracy)

avg_accuracy = np.mean(accuracy_model)
print("==> AVERAGE ACCURACY = ", avg_accuracy)
```

```
==========================================
==> AVERAGE ACCURACY =  0.9642105263157894
```

The average accuracy with subsampling/"k"=10 is 96.42%

Leave-one-out cross validation/ "LOO"

```
1  loo = LeaveOneOut()
2  loo
```

```
LeaveOneOut()
```

The Dataset split with k=n, with "n" is the total data. So, there is only 1 testing data

```
==========> LOO (Leave One Out) <==========
================================================================================
                                DATA SPLIT KE- 1
TRAIN: [  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149]
TEST: [0]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
```

This is the example of output program in experiment 1. There are 1 testing data, and 149 training data, so there will be 150 experiments.

The accuracy score of each experiment is only have 2 score, 100% or 0%. The accuracy score of experiment 1 is 100%.

The Program of KNN using Leave-one-out cross validation

```python
# LOO (Leave One Out)
print("==========> LOO (Leave One Out) <==========")
i = 0
for train_index, test_index in loo.split(x):
    print("="*80)
    i = i+1
    print("\t\t\t\t DATA SPLIT KE-", i)

    print("TRAIN:", train_index, "\nTEST:", test_index)

    x_train, x_test = x[train_index], x[test_index]
    y_train, y_test = y[train_index], y[test_index]
    #print("X_test: ", x_test)

    #KNN Classifier
    classifier = KNeighborsClassifier(n_neighbors=5)
    classifier.fit(x_train, y_train)

    #predict test data using KNN
    pred_out = classifier.predict(x_test)
    print("==> PREDICTION <==")
    #print(pred_out)
    print("-"*80)
    print("==> REAL OUTPUT <==")
    #print(y_test)
    print("-"*80)
    print("==> ACCURACY SCORE <==")
    accuracy = accuracy_score(y_test, pred_out)
    accuracy_model.append(accuracy)
    print(accuracy)
    print("="*80)
    accuracy_model.append(accuracy)

avg_accuracy = np.mean(accuracy_model)
print("==> AVERAGE ACCURACY = ", avg_accuracy)
```

```
==========> LOO (Leave One Out) <==========
================================================================================
                 DATA SPLIT KE- 1
TRAIN: [  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [0]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
================================================================================
                 DATA SPLIT KE- 2
TRAIN: [  0   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [1]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
================================================================================
                 DATA SPLIT KE- 3
TRAIN: [  0   1   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [2]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
================================================================================
                 DATA SPLIT KE- 4
TRAIN: [  0   1   2   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [3]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
================================================================================
                 DATA SPLIT KE- 5
TRAIN: [  0   1   2   3   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [4]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
================================================================================
                 DATA SPLIT KE- 6
TRAIN: [  0   1   2   3   4   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [5]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
================================================================================
                 DATA SPLIT KE- 7
TRAIN: [  0   1   2   3   4   5   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [6]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
================================================================================
                 DATA SPLIT KE- 8
TRAIN: [  0   1   2   3   4   5   6   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [7]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
================================================================================
                 DATA SPLIT KE- 9
TRAIN: [  0   1   2   3   4   5   6   7   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [8]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
================================================================================
                 DATA SPLIT KE- 10
TRAIN: [  0   1   2   3   4   5   6   7   8  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
 145 146 147 148 149] 
TEST: [9]
==> PREDICTION <==
--------------------------------------------------------------------------------
==> REAL OUTPUT <==
--------------------------------------------------------------------------------
==> ACCURACY SCORE <==
1.0
================================================================================
==> AVERAGE ACCURACY =  0.9663777089783283
```

The average accuracy with subsampling/"k"=10 is 96.63%