Diagnostic Prediction of Diabetes with Ensemble methods

September 29, 2022

1 DIAGNOSTIC PREDICTION OF DIABETES AMONG PIMA INDIAN HERITAGE

[19]:
```
import pandas
from sklearn import model_selection
from sklearn.ensemble import AdaBoostClassifier
```

[20]:
```
dataframe = pandas.read_csv('pima-indians-diabetes.csv')
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
seed = 7
num_trees = 30
```

2 Applying Boosting to Classification Models

[21]:
```
#Build Classifiers with Adaboost
#Adaboost classifiers use decision trees classifiers as default
#Passed the model within a cross validation function to evaluate results with a␣
 ↪cv technique(train data splitted into 10 consecutive folds),
#Each fold will be used as evaluation set while remaining 9 fold used as␣
 ↪training set
kfold = model_selection.KFold(n_splits=10,shuffle=True, random_state=seed)
model = AdaBoostClassifier(n_estimators=num_trees)
results = model_selection.cross_val_score(model,X,Y,cv=kfold)
print(results)
print(results.mean())
#accuracy of 76% with adaboost
```

```
[0.76623377 0.71428571 0.71428571 0.79220779 0.79220779 0.74025974
 0.68831169 0.77922078 0.80263158 0.76315789]
0.7552802460697198
```

[22]:
```
#import sys
#!{sys.executable} -m pip install xgboost
```

[24]:

```python
# Build Classifiers with SVM and XGboost Classifier
from sklearn import svm
from xgboost import XGBClassifier
clf = XGBClassifier()

seed=7
num_trees=30

kfold = model_selection.KFold(n_splits=10)
model = XGBClassifier(n_estimators = num_trees, random_state=seed, shuffle=True)
results = model_selection.cross_val_score(model, X, Y, cv=kfold)
print(results)
print(results.mean())
```

```
[17:42:53] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
cases.


[17:42:53] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
cases.


[17:42:53] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
```

```
cases.


[17:42:53] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
cases.


[17:42:54] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
cases.


[17:42:54] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
cases.


[17:42:54] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
```

```
cases.


[17:42:54] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
cases.


[17:42:54] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
cases.


[17:42:54] WARNING: C:/Users/administrator/workspace/xgboost-
win64_release_1.6.0/src/learner.cc:627:
Parameters: { "shuffle" } might not be used.

  This could be a false alarm, with some parameters getting used by language
bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually
being used
  but getting flagged wrongly here. Please open an issue if you find any such
cases.


0.7499487354750513
```

3 Random Forest with C-V splitting technique

[25]:

```python
from sklearn.datasets import load_iris
```

[26]:

```
iris_data = load_iris()
#print(iris_data)
#data_input = iris_data.data #extract input data
#data_output = iris_data.target #extract target data
```

[18]:

```
#print(data_input)
#print(data_output)
```

[27]:

```
from sklearn.model_selection import KFold
kf = KFold(n_splits=5, shuffle=True)
```

[28]:

```
print("Train Set           Test Set      ")
for train_set, test_set in kf.split(data_input):
    print(train_set, test_set)
```

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

[15]:

```
#intialize Random forest Classifier

from sklearn.ensemble import RandomForestClassifier
rf_class = RandomForestClassifier(n_estimators=10)
```

[16]:

```
#import cross validation score library from sklearn
from sklearn.model_selection import cross_val_score
print(cross_val_score(rf_class, data_input, data_output,␣
 ↪scoring='accuracy',cv=10))
```

```
[1.         0.93333333 1.         0.93333333 0.93333333 0.93333333
 0.93333333 1.         1.         1.        ]
```

[17]:

```
accuracy = cross_val_score(rf_class, data_input, data_output,␣
 ↪scoring='accuracy', cv=10).mean()*100
print('Accuracy of Random Forest is:',accuracy)
```

```
Accuracy of Random Forest is: 95.33333333333334
```