Project 2 - Classification Problem

Two of the most widely statistic I methods for analyzing categorical outcome variables linear discriminant analysis and logistic regression. For this project, I selected Linear discriminant analysis and logistic regression. Also, random sampling and stratified K-fold sampling is used for both the model. Here, Accuracy of both the models using both sampling techniques are compared. Accuracy is the proportion of correct predictions over total predictions.

Python code:

```
"""
Linear Discriminant Analysis with Training & Test Sampling, Stratified K-Fold Sampling
"""
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import (train_test_split)
from sklearn.model_selection import RepeatedStratifiedKFold

df=pd.read_csv('h2Bank.csv', header=None) # use it when file does not have headers

# Rename column titles
df.columns = ['v1', 'v2', 'v3', 'v4', 'v5', 'decision']
print (df.head()) # see first six rows to check everything

# Define independent variables and class variables
X = df[['v1', 'v2', 'v3', 'v4', 'v5']]
y = df['decision']

# split dataset into training and testing 70-30 ratio
```

```
X_train, X_test, y_train, y_test=train_test_split (X,y, test_size=0.3)# add fourth parameter random_state=10 for seeded random number generation
print('size of test dataset:',len(X_test), ' size of training dataset: ', len(X_train))

#Fit the LDA model
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
model = LinearDiscriminantAnalysis()
model.fit(X_train, y_train) #learn Discriminant Function

# Print Training and Test Accuracies
result1 = model.score(X_train, y_train)
print(("LDA Training Accuracy: %.2f%%") % (result1*100))

result = model.score(X_test, y_test)
print(("LDA Test Accuracy: %.2f%%") % (result*100.0))

from sklearn.metrics import confusion_matrix
#Create Training and Test Dataset
X_train, X_test, y_train, y_test=train_test_split(X,y,random_state=1)
y_pred=model.predict(X_test) # predict test dataset
confusion=confusion_matrix(y_test,y_pred)#,labels=[0,1])
print(confusion) # Column is Actual and Row Title is Predicted

# Cross Validation using 10 fold sampling
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

model1 = LinearDiscriminantAnalysis()
model1.fit(X, y)
```

```
#evaluate model
from sklearn.model_selection import cross_val_score
scores = cross_val_score(model1, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
print(("Stratified KFold Accuracy: %.2f%%") % (np.mean(scores)*100))
```

Output:

```
      v1     v2     v3     v4     v5  decision
0  2.463  0.056  1.682 -0.108  1.173         1
1  1.234  0.091  2.107  0.046  0.144         1
2  2.852  0.056  1.782  0.107  0.668         1
3 -0.775 -0.043  1.372  0.026  0.543         1
4  1.573  0.047  1.377  0.177  1.797         1
size of test dataset: 30  size of training dataset:  70
LDA Training Accuracy: 65.71%
LDA Test Accuracy: 60.00%
[[7 8]
 [7 3]]
Stratified KFold Accuracy: 65.67%
```

```
"""
Logistic Regression with Training & Test Sampling, Stratified KFold Sampling
"""
import pandas as pd
import numpy as np
from sklearn.model_selection import (train_test_split)
from sklearn.model_selection import RepeatedStratifiedKFold

df=pd.read_csv('h2Bank.csv', header=None) # use it when file does not have headers
# Rename column titles
```

```
df.columns = ['v1', 'v2', 'v3', 'v4', 'v5', 'decision']
print (df.head()) # see first six rows to check everything

# Define independent variables and class variables
X = df[['v1', 'v2', 'v3', 'v4', 'v5']]
y = df['decision']

# split dataset into training and testing 70-30 ratio
X_train, X_test, y_train, y_test=train_test_split (X,y, test_size=0.3)# add fourth parameter random_state=10 for seeded random number generation
print('size of test dataset:',len(X_test), ' size of training dataset: ', len(X_train))

# Logistic Regression with Area Under the Curve-may not work with Iris due to dataset loading
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression(solver="liblinear", random_state=0).fit(X_train, y_train)
from sklearn.metrics import roc_auc_score
print("ROC-AUC score: %.3f" % roc_auc_score(y, clf.predict_proba(X)[:, 1]))

# Print Training and Test Accuracies
result1 = clf.score(X_train, y_train)
print(("LR Training Accuracy: %.2f%%") % (result1*100))
result = clf.score(X_test, y_test)
print(("LR Test Accuracy: %.2f%%") % (result*100))

# Cross Validation using 10 fold sampling
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

model = LogisticRegression(solver="liblinear", random_state=0)
model.fit(X, y)
```

```
#evaluate model

from sklearn.model_selection import cross_val_score

scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)

print(("Stratified KFold Accuracy: %.2f%%") % (np.mean(scores)*100))
```

Output:

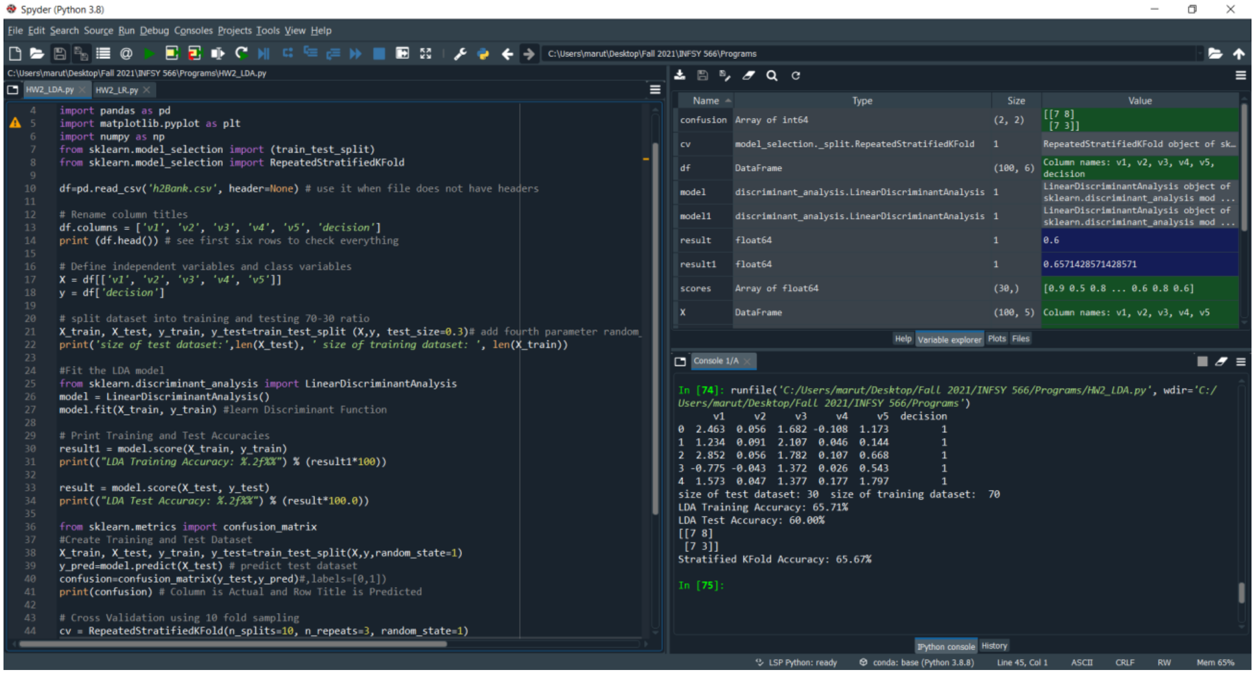
```
     v1     v2     v3     v4     v5  decision

0  2.463  0.056  1.682 -0.108  1.173         1

1  1.234  0.091  2.107  0.046  0.144         1

2  2.852  0.056  1.782  0.107  0.668         1

3 -0.775 -0.043  1.372  0.026  0.543         1

4  1.573  0.047  1.377  0.177  1.797         1

size of test dataset: 30  size of training dataset:  70

ROC-AUC score: 0.746

LR Training Accuracy: 75.71%

LR Test Accuracy: 63.33%

Stratified KFold Accuracy: 67.00%
```

Overall Result:

Accuracy %	Random Sampling	Stratified K-fold Sampling
Linear Discriminant Analysis	60.00%	65.67%
Logistic Regression	63.33%	67.00%

Comparing the results of both linear discriminant analysis and logistic regression, it is evident that logistic regression using stratified K-fold sampling yields better accuracy compared to linear discriminant analysis. While both are appropriate for the development of linear classification models, linear discriminant analysis makes more assumptions about the underlying data. Hence, it is assumed that logistic regression is the more flexible and more robust method. Hence, I consider Logistic regression is better compared to LDA. Similarly, stratified Kfold sampling is better compared to random sampling.

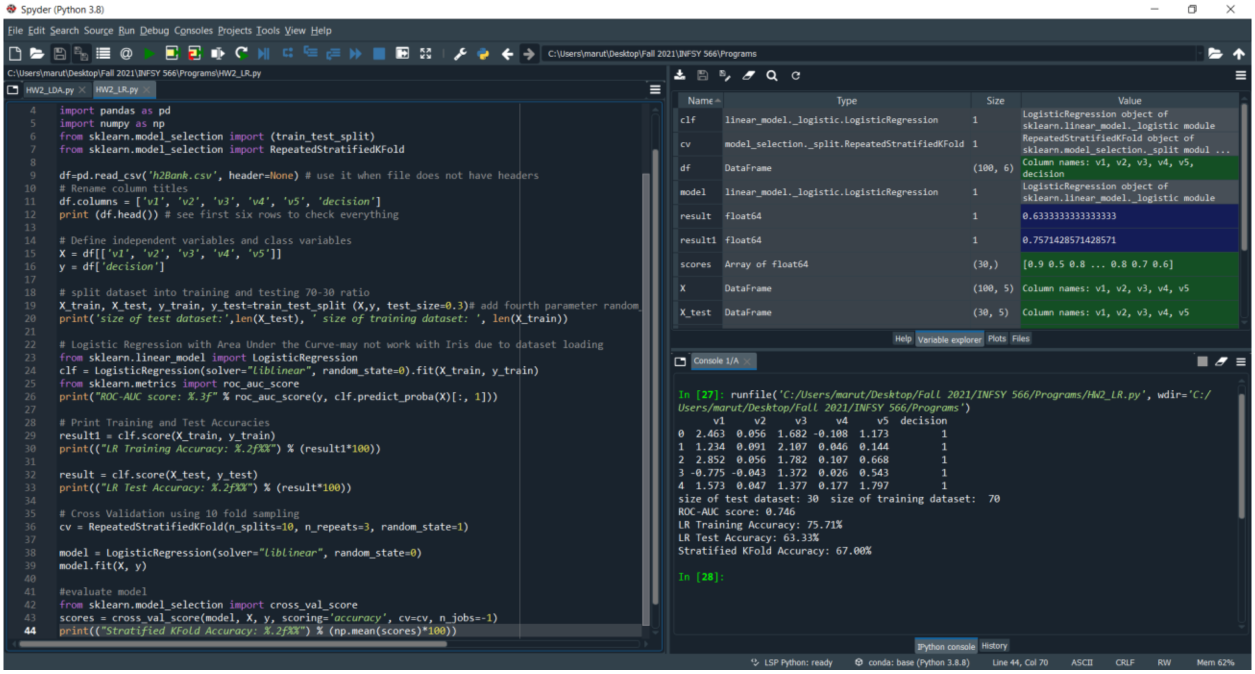
Screenshots:


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\marut\Desktop\Fall 2021\INFSY 566\Programs
C:\Users\marut\Desktop\Fall 2021\INFSY 566\Programs\HW2_LDA.py
HW2_LDA.py
HW2_LR.py
Help
Variable explorer
Plots
Files
Console 1/A
IPython console
History
LSP Python: ready
conda: base (Python 3.8.8)
Line 45, Col 1
ASCII
CRLF
RW
Mem 65%

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import (train_test_split)
from sklearn.model_selection import RepeatedStratifiedKFold

df=pd.read_csv('h2Bank.csv', header=None) # use it when file does not have headers

# Rename column titles
df.columns = ['v1', 'v2', 'v3', 'v4', 'v5', 'decision']
print (df.head()) # see first six rows to check everything

# Define independent variables and class variables
X = df[['v1', 'v2', 'v3', 'v4', 'v5']]
y = df['decision']

# split dataset into training and testing 70-30 ratio
X_train, X_test, y_train, y_test=train_test_split (X,y, test_size=0.3)# add fourth parameter random_
print('size of test dataset:',len(X_test), ' size of training dataset: ', len(X_train))

#Fit the LDA model
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
model = LinearDiscriminantAnalysis()
model.fit(X_train, y_train) #learn Discriminant Function

# Print Training and Test Accuracies
result1 = model.score(X_train, y_train)
print(("LDA Training Accuracy: %.2f%%") % (result1*100))

result = model.score(X_test, y_test)
print(("LDA Test Accuracy: %.2f%%") % (result*100.0))

from sklearn.metrics import confusion_matrix
#Create Training and Test Dataset
X_train, X_test, y_train, y_test=train_test_split(X,y,random_state=1)
y_pred=model.predict(X_test) # predict test dataset
confusion=confusion_matrix(y_test,y_pred)#,labels=[0,1])
print(confusion) # Column is Actual and Row Title is Predicted

# Cross Validation using 10 fold sampling
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
```

Name	Type	Size	Value
confusion	Array of int64	(2, 2)	[[7 8] [7 3]]
cv	model_selection._split.RepeatedStratifiedKFold	1	RepeatedStratifiedKFold object of sk…
df	DataFrame	(100, 6)	Column names: v1, v2, v3, v4, v5, decision
model	discriminant_analysis.LinearDiscriminantAnalysis	1	LinearDiscriminantAnalysis object of sklearn.discriminant_analysis mod ...
model1	discriminant_analysis.LinearDiscriminantAnalysis	1	LinearDiscriminantAnalysis object of sklearn.discriminant_analysis mod ...
result	float64	1	0.6
result1	float64	1	0.6571428571428571
scores	Array of float64	(30,)	[0.9 0.5 0.8 ... 0.6 0.8 0.6]
X	DataFrame	(100, 5)	Column names: v1, v2, v3, v4, v5

```
In [74]: runfile('C:/Users/marut/Desktop/Fall 2021/INFSY 566/Programs/HW2_LDA.py', wdir='C:/
Users/marut/Desktop/Fall 2021/INFSY 566/Programs')
      v1     v2     v3     v4     v5  decision
0  2.463  0.056  1.682 -0.108  1.173         1
1  1.234  0.091  2.107  0.046  0.144         1
2  2.852  0.056  1.782  0.107  0.668         1
3 -0.775 -0.043  1.372  0.026  0.543         1
4  1.573  0.047  1.377  0.177  1.797         1
size of test dataset: 30  size of training dataset:  70
LDA Training Accuracy: 65.71%
LDA Test Accuracy: 60.00%
[[7 8]
 [7 3]]
Stratified KFold Accuracy: 65.67%

In [75]:
```


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\marut\Desktop\Fall 2021\INFSY 566\Programs
C:\Users\marut\Desktop\Fall 2021\INFSY 566\Programs\HW2_LR.py
HW2_LDA.py
HW2_LR.py
Help
Variable explorer
Plots
Files
Console 1/A
IPython console
History
LSP Python: ready
conda: base (Python 3.8.8)
Line 44, Col 70
ASCII
CRLF
RW
Mem 62%

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import (train_test_split)
from sklearn.model_selection import RepeatedStratifiedKFold

df=pd.read_csv('h2Bank.csv', header=None) # use it when file does not have headers
# Rename column titles
df.columns = ['v1', 'v2', 'v3', 'v4', 'v5', 'decision']
print (df.head()) # see first six rows to check everything

# Define independent variables and class variables
X = df[['v1', 'v2', 'v3', 'v4', 'v5']]
y = df['decision']

# split dataset into training and testing 70-30 ratio
X_train, X_test, y_train, y_test=train_test_split (X,y, test_size=0.3)# add fourth parameter random_
print('size of test dataset:',len(X_test), ' size of training dataset: ', len(X_train))

# Logistic Regression with Area Under the Curve-may not work with Iris due to dataset loading
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression(solver="liblinear", random_state=0).fit(X_train, y_train)
from sklearn.metrics import roc_auc_score
print("ROC-AUC score: %.3f" % roc_auc_score(y, clf.predict_proba(X)[:, 1]))

# Print Training and Test Accuracies
result1 = clf.score(X_train, y_train)
print(("LR Training Accuracy: %.2f%%") % (result1*100))

result = clf.score(X_test, y_test)
print(("LR Test Accuracy: %.2f%%") % (result*100))

# Cross Validation using 10 fold sampling
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

model = LogisticRegression(solver="liblinear", random_state=0)
model.fit(X, y)

#evaluate model
from sklearn.model_selection import cross_val_score
scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
print(("Stratified KFold Accuracy: %.2f%%") % (np.mean(scores)*100))
```

Name	Type	Size	Value
clf	linear_model._logistic.LogisticRegression	1	LogisticRegression object of sklearn.linear_model._logistic module
cv	model_selection._split.RepeatedStratifiedKFold	1	RepeatedStratifiedKFold object of sklearn.model_selection._split modul ...
df	DataFrame	(100, 6)	Column names: v1, v2, v3, v4, v5, decision
model	linear_model._logistic.LogisticRegression	1	LogisticRegression object of sklearn.linear_model._logistic module
result	float64	1	0.6333333333333333
result1	float64	1	0.7571428571428571
scores	Array of float64	(30,)	[0.9 0.5 0.8 ... 0.8 0.7 0.6]
X	DataFrame	(100, 5)	Column names: v1, v2, v3, v4, v5
X_test	DataFrame	(30, 5)	Column names: v1, v2, v3, v4, v5

```
In [27]: runfile('C:/Users/marut/Desktop/Fall 2021/INFSY 566/Programs/HW2_LR.py', wdir='C:/
Users/marut/Desktop/Fall 2021/INFSY 566/Programs')
      v1     v2     v3     v4     v5  decision
0  2.463  0.056  1.682 -0.108  1.173         1
1  1.234  0.091  2.107  0.046  0.144         1
2  2.852  0.056  1.782  0.107  0.668         1
3 -0.775 -0.043  1.372  0.026  0.543         1
4  1.573  0.047  1.377  0.177  1.797         1
size of test dataset: 30  size of training dataset:  70
ROC-AUC score: 0.746
LR Training Accuracy: 75.71%
LR Test Accuracy: 63.33%
Stratified KFold Accuracy: 67.00%

In [28]:
```