Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh

Performance Evaluation of Public Bus Service Quality Using Machine Learning

-Software: Python jupyter notebook

-Libraries: pandas, matplotlib, sklearn

-Survey collection method: Questionnaire survey (QS) based on Stated Preference (SP)

-Feature/attributes: 5 (Cleanness, frequency, fare, behavior, ventilation) and output (Overall)

-Scale: Likert scale (1-5), 1: very poor, 2: poor, 3: satisfactory, 4: fair, 5: Excellent

-Survey data: 50

-Excel data file: SQ.csv

-Machine Learning (ML) Algorithms: Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Classification and Regression Trees (CART), Gaussian Naive Bayes (NB), Support Vector Machines (SVM).

Code:

1. Import libraries

```
# Load libraries
import pandas
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
```

Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

2. Load Dataset

```
# Load dataset
url = "/Users/Administrator/Desktop/python practice/SQ.csv"
names = ['Cleaness','frequency','Fare','behavior','Ventilation','Overall']

dataset = pandas.read_csv(url, names=names)
```

3. Summarize the Dataset

3.1 Dimensions of Dataset

```
# shape
print(dataset.shape)
```

Output:

```
(50, 6)
```

3.2 Peek at the Data

```
# head
```

Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh

print(dataset.head(20))

Output:

	Cleaness	frequency	Fare	behavior	Ventilation	Overall
0	1	2	1	1	1	1
1	1	1	1	1	1	1
2	1	2	1	2	2	1
3	2	2	1	3	2	1
4	1	1	2	2	1	1
5	3	3	2	3	2	2
6	4	2	1	2	2	2
7	3	3	2	3	3	2
8	2	1	2	1	2	2
9	1	1	2	2	3	2
10	2	2	3	3	2	3
11	4	2	3	3	2	3
12	2	3	2	3	2	3
13	2	3	3	2	3	3
14	1	4	2	4	2	4
15	2	3	4	4	3	4
16	3	3	4	3	3	4
17	3	3	2	4	2	4
18	2	1	2	2	4	4
19	5	3	4	3	2	5

3.3 Statistical Summary

descriptions

print(dataset.describe()

output:

	Cleaness	frequency	Fare	behavior	Ventilation	Overall
count	50.000000	50.000000	50.000000	50.000000	50.000000	50.000000
mean	2.760000	2.440000	2.040000	2.640000	2.420000	2.760000
std	1.436549	0.704504	1.009344	0.875051	0.927802	1.519398
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	2.000000	2.000000	1.000000	2.000000	2.000000	1.000000
50%	3.000000	3.000000	2.000000	3.000000	2.000000	2.000000
75%	4.000000	3.000000	3.000000	3.000000	3.000000	4.000000
max	5.000000	4.000000	5.000000	4.000000	4.000000	5.000000

3.4 Class Distribution

class distribution

Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh

```
print(dataset.groupby('Overall').size())
```

output:

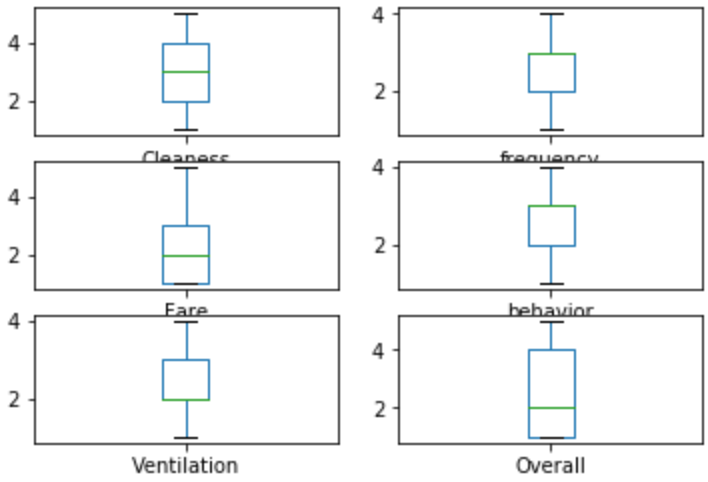
```
Overall
1    14
2    12
3     6
4     8
5    10
dtype: int64
```

4. Data Visualization

4.1 Univariate Plots

```
# box and whisker plots
dataset.plot(kind='box', subplots=True, layout=(3,2), sharex=False, sharey=False)
plt.show()
```

output: (as png)

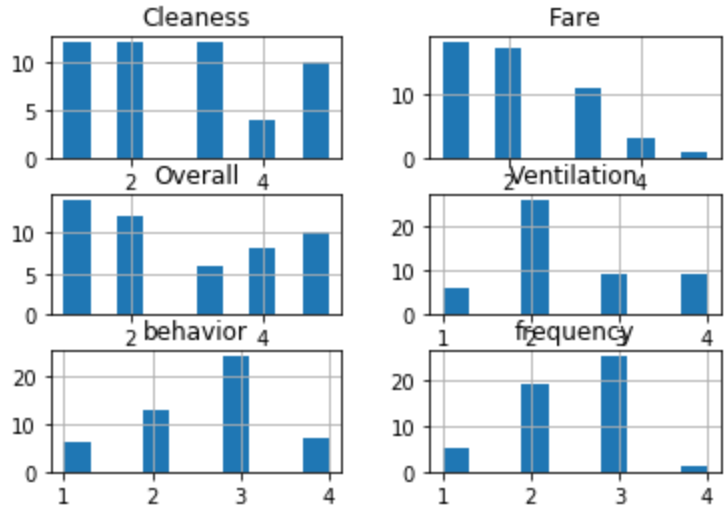



```
# histograms
```

Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh

```
dataset.hist()
plt.show()
```

Output:

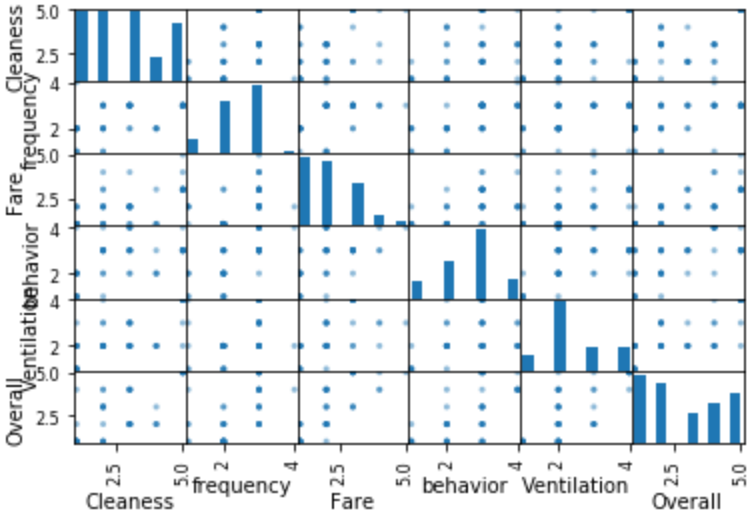


4.2 Multivariate Plots

```
# scatter plot matrix
scatter_matrix(dataset)
plt.show()
```

Output: (as png)

Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh



5. Evaluate Some Algorithms

5.1 Create a Validation Dataset

```
array = dataset.values

X = array[:,0:5]

Y = array[:,5]

validation_size = 0.20

seed = 7

X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y,
test_size=validation_size, random_state=seed)
```

5.2 Test Harness

```
# Test options and evaluation metric

seed = 7

scoring = 'accuracy'
```

Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh

5.3 Build Models

Let's evaluate 6 different algorithms:

- Logistic Regression (LR)
- Linear Discriminant Analysis (LDA)
- K-Nearest Neighbors (KNN).
- Classification and Regression Trees (CART).
- Gaussian Naive Bayes (NB).
- Support Vector Machines (SVM).

```
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
# evaluate each model in turn
results = []
names = []
for name, model in models:
kfold = model_selection.KFold(n_splits=10, random_state=seed)
cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
results.append(cv_results)
names.append(name)
msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
print(msg)
```

Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh

5.4 Select Best Model

We now have 6 models and accuracy estimations for each. We need to compare the models to each other and select the most accurate.

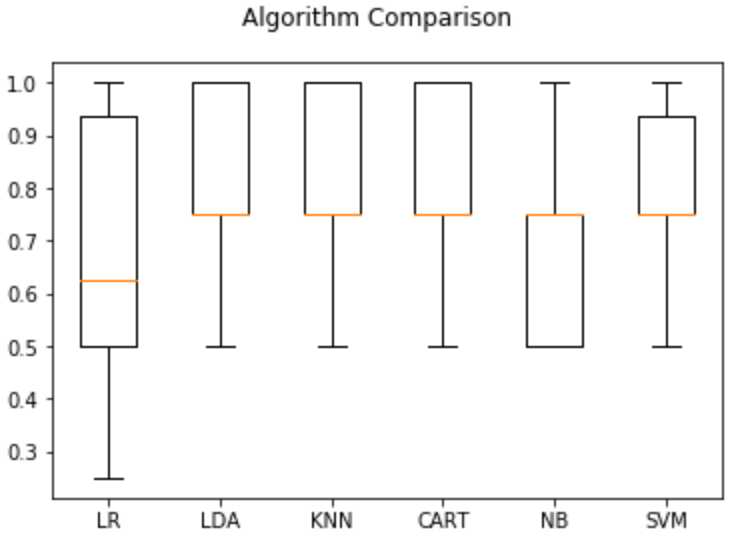
output:

```
LR: 0.675000 (0.251247)
LDA: 0.800000 (0.187083)
KNN: 0.800000 (0.187083)
CART: 0.800000 (0.187083)
NB: 0.675000 (0.160078)
SVM: 0.800000 (0.150000)
```

```
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```

output:

Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh



5.5 Make Predictions

The KNN algorithm is very simple and was an accurate model based on our tests. Now we want to get an idea of the accuracy of the model on our validation set.

```
# Make predictions on validation dataset
knn = KNeighborsClassifier()
knn.fit(X_train, Y_train)
predictions = knn.predict(X_validation)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))
```

output:

```
0.9
[[4 0 0 0 0]
 [0 1 0 0 0]
 [0 1 1 0 0]
```

Saurav Barua
Assistant Professor, Department of Civil Engineering,
Daffodil International University, Dhaka, Bangladesh

```
 [0 0 0 1 0]
 [0 0 0 0 2]]
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         4
           2       0.50      1.00      0.67         1
           3       1.00      0.50      0.67         2
           4       1.00      1.00      1.00         1
           5       1.00      1.00      1.00         2

   micro avg       0.90      0.90      0.90        10
   macro avg       0.90      0.90      0.87        10
weighted avg       0.95      0.90      0.90        10
```

Reference:

Brownlee, J., February 2019, "Your First Machine Learning Project in Python Step-By-Step", Python Machine Learning.

https://machinelearningmastery.com/machine-learning-in-python-step-by-step/