Machine Learning

Dr.R.V.S.Lalitha
Professor
Department of C.S.E.
Aditya College of Engineering & Technology
Surampalem

Subject: Machine learning
Topic: Demonstrating Implementation of classification algorithms through COLAB environment
Teaching Methodology: Technology and Innovative methods of Teaching

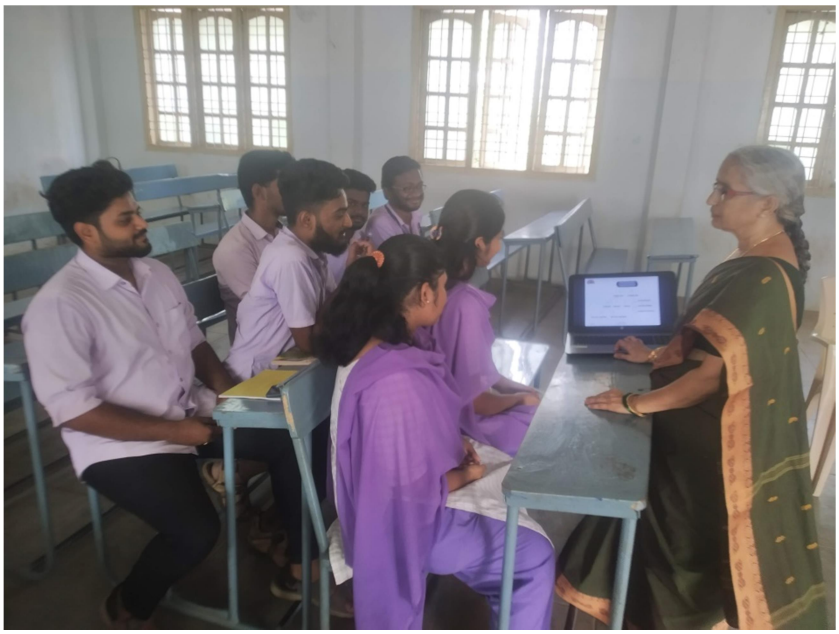


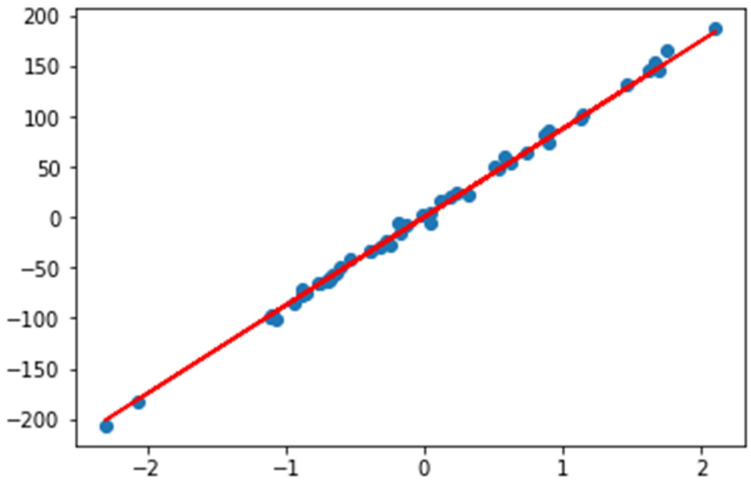
Fig.1. Understanding theory concepts through advanced code based tools (Google COLAB)

Ridge regression

https://towardsdatascience.com/ridge-regression-python-example-f015345d936b

```
from sklearn.datasets import make_regression
from matplotlib import pyplot as plt
import numpy as np
from sklearn.linear_model import Ridge
X, y, coefficients = make_regression(
    n_samples=50,
    n_features=1,
    n_informative=1,
    n_targets=1,
    noise=5,
    coef=True,
    random_state=1
)
alpha = 1
n, m = X.shape
I = np.identity(m)
w = np.dot(np.dot(np.linalg.inv(np.dot(X.T, X) + alpha * I), X.T), y)
print('w',w)
print('coefficients',coefficients)
plt.scatter(X, y)
plt.plot(X, w*X, c='red')
rr = Ridge(alpha=1)
rr.fit(X, y)
w = rr.coef_
print('w',w)
```

```
w [87.37153533]
coefficients 90.34019152878835
w [87.39928165]
```



Linear model coefficients

```
from scipy import stats
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

slope, intercept, r, p, std_err = stats.linregress(x, y)
print(r)
print(slope)
print(intercept)
print(std_err)
print(p)
#The r value ranges from -1 to 1, where 0 means no relationship, and 1 (and -1) means 100% related.
```

```
-0.758591524376155
-1.7512877115526118
103.10596026490066
0.453536157607742
0.0026468739224561064
```

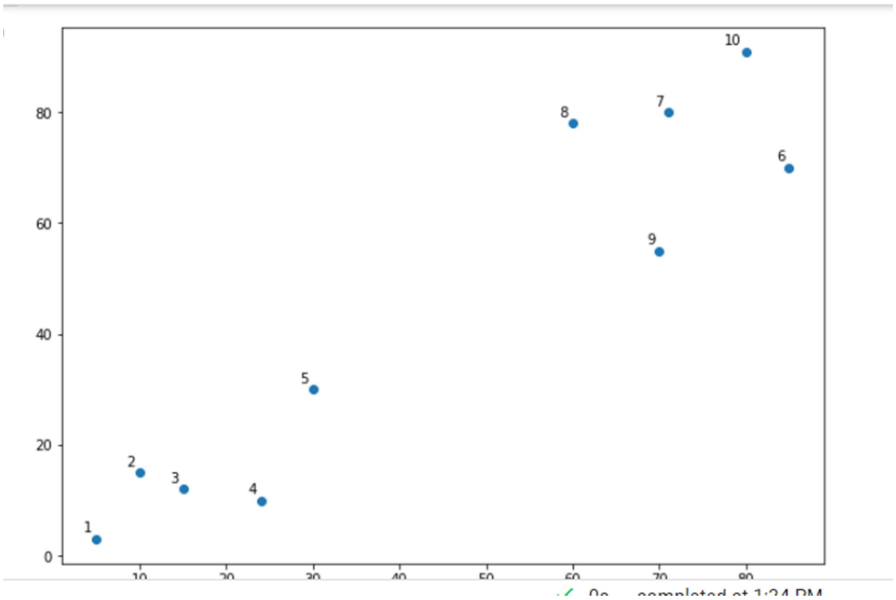
Hierarchical clustering

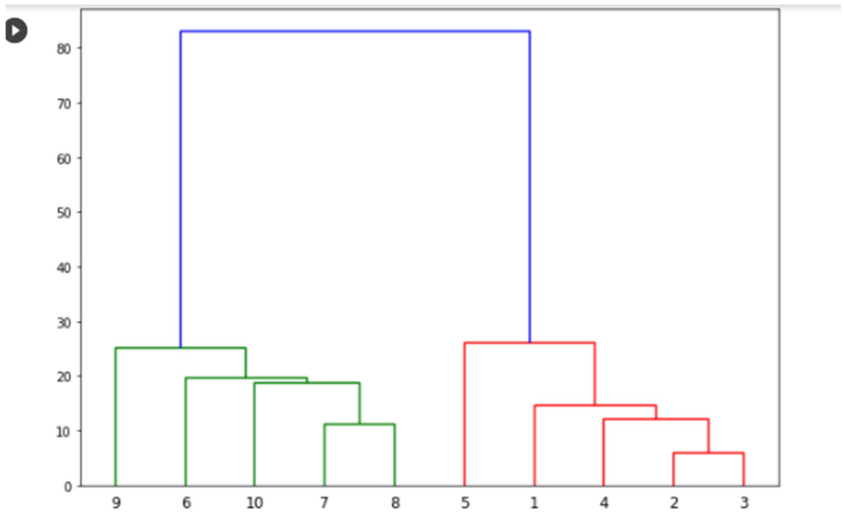
https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/#:~:text=Steps%20to%20Perform%20Hierarchical%20Clustering&text=At%20t

he%20start%2C%20treat%20each%20data%20point%20as%20one%20cluster.&text=Form%20a%20cluster%20by%20joining,one%20big%20cluster%20is%20formed.

```
import numpy as np
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

X = np.array([[5,3],
    [10,15],
    [15,12],
    [24,10],
    [30,30],
    [85,70],
    [71,80],
    [60,78],
    [70,55],
    [80,91],])
import matplotlib.pyplot as plt

labels = range(1, 11)
plt.figure(figsize=(10, 7))
plt.subplots_adjust(bottom=0.1)
plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y in zip(labels, X[:, 0], X[:, 1]):
    plt.annotate(
        label,
        xy=(x, y), xytext=(-3, 3),
        textcoords='offset points', ha='right', va='bottom')
plt.show()
linked = linkage(X, 'centroid')
labelList = range(1, 11)
```

```
plt.figure(figsize=(10, 7))
dendrogram(linked,
           orientation='top',
           labels=labelList,
           distance_sort='descending',
           show_leaf_counts=True)
plt.show()
```

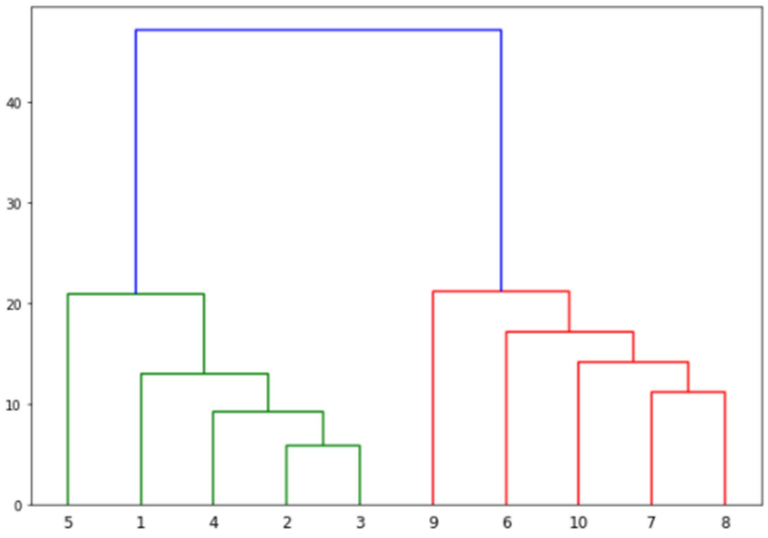





```
import numpy as np
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt
```

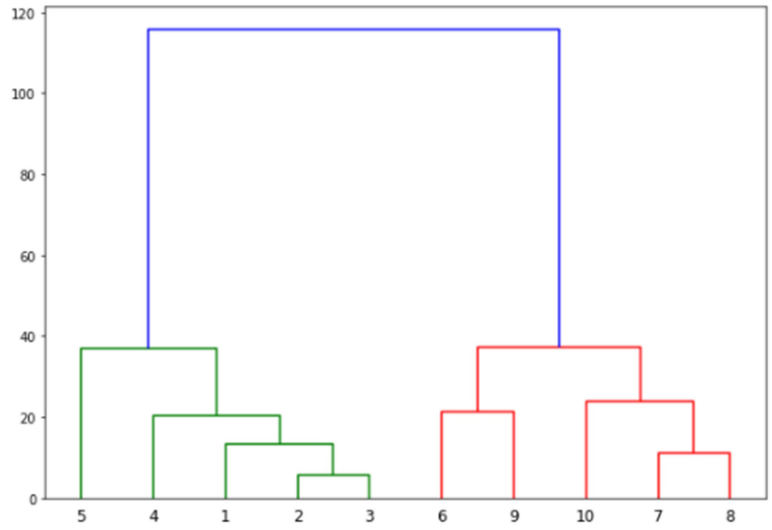
```
X = np.array([[5,3],
    [10,15],
    [15,12],
    [24,10],
    [30,30],
    [85,70],
    [71,80],
    [60,78],
    [70,55],
    [80,91],])
import matplotlib.pyplot as plt

labels = range(1, 11)
plt.figure(figsize=(10, 7))
plt.subplots_adjust(bottom=0.1)
plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y in zip(labels, X[:, 0], X[:, 1]):
    plt.annotate(
        label,
        xy=(x, y), xytext=(-3, 3),
        textcoords='offset points', ha='right', va='bottom')
plt.show()


linked = linkage(X, 'centroid')

labelList = range(1, 11)

plt.figure(figsize=(10, 7))
dendrogram(linked,
            orientation='top',
            labels=labelList,
```

```
        distance_sort='descending',
        show_leaf_counts=True)
plt.show()
```




```
import numpy as np
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

X = np.array([[5,3],
    [10,15],
    [15,12],
    [24,10],
    [30,30],
    [85,70],
    [71,80],
    [60,78],
    [70,55],
    [80,91],])
import matplotlib.pyplot as plt

labels = range(1, 11)
plt.figure(figsize=(10, 7))
plt.subplots_adjust(bottom=0.1)
```

```
plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y in zip(labels, X[:, 0], X[:, 1]):
    plt.annotate(
        label,
        xy=(x, y), xytext=(-3, 3),
        textcoords='offset points', ha='right', va='bottom')
plt.show()


linked = linkage(X, 'complete')

labelList = range(1, 11)

plt.figure(figsize=(10, 7))
dendrogram(linked,
           orientation='top',
           labels=labelList,
           distance_sort='descending',
           show_leaf_counts=True)
plt.show()
```




```
from matplotlib import pyplot as plt
```

```
X = np.array([[5,3],
    [10,15],
    [15,12],
    [24,10],
    [30,30],
    [85,70],
    [71,80],
    [60,78],
    [70,55],
    [80,91],])
import matplotlib.pyplot as plt

labels = range(1, 11)
plt.figure(figsize=(10, 7))
plt.subplots_adjust(bottom=0.1)
plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y in zip(labels, X[:, 0], X[:, 1]):
    plt.annotate(
        label,
        xy=(x, y), xytext=(-3, 3),
        textcoords='offset points', ha='right', va='bottom')
plt.show()


linked = linkage(X, 'average')

labelList = range(1, 11)

plt.figure(figsize=(10, 7))
dendrogram(linked,
            orientation='top',
            labels=labelList,
```

```
        distance_sort='descending',
        show_leaf_counts=True)
plt.show()
```