

Microsoft Excel interface showing a spreadsheet with columns A through W and rows 1 through 30. The spreadsheet contains data in columns A, B, and C, with column D labeled "NoLoad". The status bar at the bottom indicates "Energy Meter". A video call window is visible in the bottom right corner, showing a person wearing a headset.

Excel Ribbon: Clipboard, Font, Alignment, Number, Conditional Formatting, Styles, Cells, Editing.

Spreadsheet Data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	0	0	0	NoLoad																			
2	0	0	0	NoLoad																			
3	0	0	0	NoLoad																			
4	0	0	0	NoLoad																			
5	0	0	0	NoLoad																			
6	0	0	0	NoLoad																			
7	0	0	0	NoLoad																			
8	0.3833	0	0	NoLoad																			
9	0	0	0	NoLoad																			
10	0	0	0	NoLoad																			
11	0	0	0	NoLoad																			
12	0	0	0	NoLoad																			
13	0	0	0	NoLoad																			
14	0	0	0	NoLoad																			
15	0	0	0	NoLoad																			
16	0	0	0	NoLoad																			
17	0	0	0	NoLoad																			
18	0	0	0	NoLoad																			
19	0	0	0	NoLoad																			
20	0	0	0	NoLoad																			
21	0.3833	0	0	NoLoad																			
22	0.3833	0	0	NoLoad																			
23	0.3833	0	0	NoLoad																			
24	0	0	0	NoLoad																			
25	0	0	0	NoLoad																			
26	0	0	0	NoLoad																			
27	0	0	0	NoLoad																			
28	0	0	0	NoLoad																			
29	0	0	0	NoLoad																			
30	0	0	0	NoLoad																			

Taskbar: Windows Start button, Search bar, Task View, File Explorer, Microsoft Edge, Google Chrome, Zoom, Excel, System tray (29°C Rain).

EvaluatingVariousMLModelforEn x Training&TestingMLAlgorithmFo x Day-10\_BreastCancerDetection\_V x +

colab.research.google.com/drive/1V5Chq8qwuk4jOLO19bh7hiSwArzoU\_4y#scrollTo=-7zM7CWgwHT-

CO EvaluatingVariousMLModelforEnergyMeter.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share

RAM Disk

Editing

+ Code + Text

RAM Disk

Editing

Loading Dataset

from pandas import read\_csv, #handling csv

[ ] from google.colab import files  
uploaded = files.upload()

[ ] fileName = "Energy Meter.csv"  
names = ['Voltage', 'Current', 'Power', 'class']  
dataset = read\_csv(fileName, names=names)  
dataset

0s completed at 19:17

EvaluatingVariousMLModelforEn x Training&TestingMLAlgorithmFo x Day-10\_BreastCancerDetection\_v x +

colab.research.google.com/drive/1V5Chq8qwuk4jOLO19bh7hiSwArzoU\_4y#scrollTo=dY7V1TXywhT\_

EvaluatingVariousMLModelforEnergyMeter.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

400 rows × 4 columns

RAM Disk Editing

Summarize Dataset

```
print(dataset.shape) #no. of rows and columns
print(dataset.head(5)) #Top 5 values of Dataset
print(dataset.describe()) #Details of Dataset
print(dataset.groupby('class').size()) #Count of data based on the classes
```

Visualize Data

```
[ ] from pandas.plotting import scatter_matrix
    from matplotlib import pyplot
```

0s completed at 19:17

EvaluatingVariousMLModelforEn x Training&TestingMLAlgorithmFo x Day-10\_BreastCancerDetection\_V x +

colab.research.google.com/drive/1V5Chq8qwuk4jOLO19bh7hiSwArzoU\_4y#scrollTo=90vKuZX5wHUH

EvaluatingVariousMLModelforEnergyMeter.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

Editing

```
[5] from pandas.plotting import scatter_matrix
from matplotlib import pyplot

dataset.plot(kind='bar', subplots=True, layout=(2,2))
pyplot.title('BAR PLOT')
pyplot.show()

dataset.hist()
pyplot.title('HISTOGRAM PLOT')
pyplot.show()

#Scatter_matrix library
scatter_matrix(dataset)
pyplot.title('SCATTER PLOT')
pyplot.show()
```

+ Code + Text

RAM Disk Editing

## Evaluating various ML Algorithm

```
# 6 ML Algorithm
#!pip install scikit-learn
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

from sklearn.model_selection import train_test_split #splitting dataset into train & Test

from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
```

✓ 10s completed at 19:20



EvaluatingVariousMLModelforEnergyMeter.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

Editing


```
[8] from sklearn.model_selection import StratifiedKFold
```

```
array = dataset.values
X = array[:,0:3]
y = array[:,3]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=0.20, random_state=1, shuffle=True)
```

```
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')))
```

```
[ ] results = []
names = []
```

0s completed at 19:23



+ Code + Text

RAM Disk Editing

```
models.append(('NB', GaussianNB()))  
models.append(('SVM', SVC(gamma='auto')))
```

```
results = []  
names = []  
res = []  
for name, model in models:  
    kfold = StratifiedKFold(n_splits=10, random_state=None)  
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')  
    results.append(cv_results)  
    names.append(name)  
    res.append(cv_results.mean())  
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))  
  
plt.ylim(.990, .999)  
plt.bar(names, res, color='maroon', width = 0.6)  
  
plt.title('Algorithm Comparison')  
plt.show()
```

0s completed at 19:23





CO

EvaluatingVariousMLModelforEnergyMeter.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment

Share

Settings

Profile

+ Code + Text

RAM  Disk  Editing

0.5

▶

```
pyplot.title('Algorithm Comparison')
pyplot.show()
```

LR: 0.996875 (0.009375)

LDA: 0.996875 (0.009375)

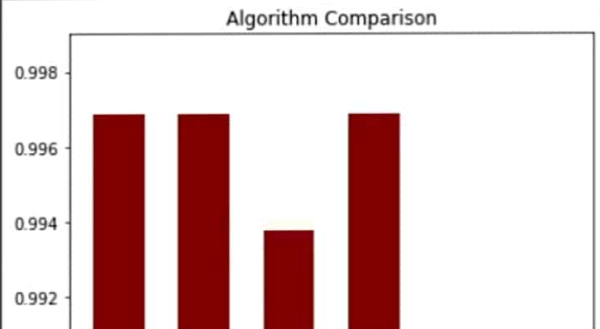
KNN: 0.993750 (0.018750)

CART: 0.996875 (0.009375)

NB: 0.990625 (0.020010)

SVM: 0.990625 (0.020010)

Algorithm Comparison



Algorithm	Accuracy	Std Dev
LR	0.996875	0.009375
LDA	0.996875	0.009375
KNN	0.993750	0.018750
CART	0.996875	0.009375







+ Code + Text



RAM



Disk



Editing



0s

```
from pandas import read_csv
from sklearn.model_selection import train_test_split

from sklearn.svm import SVC
```

```
[ ] from google.colab import files
    uploaded = files.upload()
```

Choose Files

No file chosen

Upload widget is only available when the

executed in the current browser session. Please rerun this cell to enable.

Saving Energy Meter.csv to Energy Meter.csv



0s

completed at 19:34





Training&TestingMLAlgorithmForEnerg...

File Edit View Insert Runtime Tools Help



Comment



Share



+ Code + Text



RAM  
Disk



Editing



~~from~~ google.colab import files

[2] uploaded = files.upload()

Choose Files

Energy Meter.csv

• **Energy Meter.csv**(application/vnd.ms-excel) - 16697 bytes, last modified: 1/27/2020 - 100% done  
Saving Energy Meter.csv to Energy Meter.csv



url = "Energy Meter.csv"

names = ['Voltage', 'Current', 'Power', 'class']

dataset = read\_csv(url, names=names)

[ ] array = dataset.values



5s

completed at 19:35





+ Code + Text



RAM



Disk



Editing



0s

```
X = array[:,0:3]
[4] y = array[:,3]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=0.5
model = SVC(gamma='auto')
```



0s

```
model.fit(X_train, Y_train)
```



```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

```
[ 1] import pickle
```



0s

completed at 19:35





+ Code + Text



RAM  
Disk



Editing



```
max_iter=1, probability=False, random_state=None, shrinkage=True,  
tol=0.001, verbose=False)
```



0s



```
import pickle  
filename = 'model.pkl'  
pickle.dump(model, open(filename, 'wb'))
```

```
[ ] loaded_model = pickle.load(open(filename, 'rb'))  
result = loaded_model.score(X_validation, Y_validation)  
print(result)
```

0.985



0s

completed at 19:37





## Training&TestingMLAlgorithmForEner...

File Edit View Insert Runtime Tools Help All c...



Comment



Share



+ Code + Text



RAM  
Disk



Editing



0.985



```
value = [[212.713,0.176,37.65]]  
predictions = model.predict(value)  
print(predictions[0])
```



Medium



0s

completed at 19:38



model.pkl

Type here to search



29°C Rain







Day-10\_BreastCancerDetection\_Variou...

File Edit View Insert Runtime Tools Help Last...



Comment



Share



+ Code + Text

Connect

Editing

## Day-10\_BreastCancerDetection\_VariousMLAlgorithm

### Importing Libraries

```
[ ] import pandas as pd #useful for loading the dataset
import numpy as np #to perform array
```



model.pkl

Type here to search



29°C Rain



```
[ ] from matplotlib import pyplot
```

Choose Dataset from Local Directory

```
[ ] from google.colab import files  
uploaded = files.upload()
```

Load Dataset





+ Code + Text

Connect

Editing

```
[ ] dataset = pd.read_csv('data.csv')
```

### Summarize Dataset

```
[ ] print(dataset.shape)
    print(dataset.head(5))
```

### Mapping Class String Values to Numbers



model.pkl

Type here to search



29°C Rain



```
print(dataset.head(5))
```

### Mapping Class String Values to Numbers

```
[ ] dataset['diagnosis'] = dataset['diagnosis'].map({'B': 0, 'M': 1}).astype(int)
print(dataset.head)
```

### Segregate Dataset into X(Input/IndependentVariable) &





+ Code + Text

Connect

Editing

```
[ ] X = dataset.iloc[:, 2:32].values
X
```

```
[ ] Y = dataset.iloc[:,1].values
Y
```

Splitting Dataset into Train & Test

model.pkl

Type here to search



29°C Rain



```
[ ] Y
```

## Splitting Dataset into Train & Test

```
[ ] from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.25, random_
```

## Feature Scaling





+ Code + Text

Connect

Editing

model

```
[ ] from sklearn.preprocessing import StandardScaler
    sc = StandardScaler()
    X_train = sc.fit_transform(X_train)
    X_test = sc.transform(X_test)
```

Validating some ML algorithm by its accuracy - Model Score



model.pkl

Type here to search



29°C Rain



Connect

Editing

```
[ ] from sklearn.naive_bayes import GaussianNB
    from sklearn.svm import SVC

    from sklearn.model_selection import cross_val_score
    from sklearn.model_selection import StratifiedKFold
```

```
[ ] models = []
    models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
    models.append(('LDA', LinearDiscriminantAnalysis()))
    models.append(('KNN', KNeighborsClassifier()))
    models.append(('CART', DecisionTreeClassifier()))
```



model.pkl

Type here to search



colab.research.google.com/drive/1XMLooXzqhsTbj-KgmxS0yIM3jn284IFB

# Day-10\_BreastCancerDetection\_Variou...

File Edit View Insert Runtime Tools Help Last...

+ Code + Text Connect Editing

```
[ ] from sklearn.model_selection import StratifiedKFold


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')))
```

[ ] results = []

model.pkl

Type here to search

29°C Rain





Connect

Editing

```
[ ] kfold = StratifiedKFold(n_splits=10, random_state=None)
cv_results = cross_val_score(model, X_train, y_train, cv=kfold, scoring='accuracy')
results.append(cv_results)
names.append(name)
res.append(cv_results.mean())
print('%s: %f' % (name, cv_results.mean()))

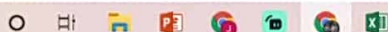
pyplot.ylim(.900, .999)
pyplot.bar(names, res, color='maroon', width = 0.6)

pyplot.title('Algorithm Comparison')
pyplot.show()
```



model.pkl

Type here to search



29°C Rain



+ Code + Text

Connect

Editing

```
[ ] pyplot.show()
```

### ▼ Training & Prediction using the algorithm with high accuracy

```
[ ] from sklearn.svm import SVC
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1))
```



model.pkl

Type here to search



29°C Rain