```
import pandas as pd
```

```
from google.colab import files
```

```
uploaded = files.upload()
```

Choose Files headbrain.csv

- **headbrain.csv**(application/vnd.ms-excel) - 3370 bytes, last modified: 1/24/2021 - 100% done

```
Saving headbrain.csv to headbrain (1).csv
```

```
import io
```

```
df = pd.read_csv(io.BytesIO(uploaded['headbrain.csv']))
print(df)
```

```
     Gender  Age Range  Head Size(cm^3)  Brain Weight(grams)
0         1          1             4512                 1530
1         1          1             3738                 1297
2         1          1             4261                 1335
3         1          1             3777                 1282
4         1          1             4177                 1590
..      ...        ...              ...                  ...
232       2          2             3214                 1110
233       2          2             3394                 1215
234       2          2             3233                 1104
235       2          2             3352                 1170
236       2          2             3391                 1120

[237 rows x 4 columns]
```

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
```

```
from sklearn import datasets
```

```
df = datasets.load_iris()
```

```
df
```

```
             [6.3, 2.9, 5.6, 1.8],
             [6.5, 3. , 5.8, 2.2],
             [7.6, 3. , 6.6, 2.1],
             [4.9, 2.5, 4.5, 1.7],
             [7.3, 2.9, 6.3, 1.8],
             [6.7, 2.5, 5.8, 1.8],
             [6.7, 2.5, 5.8, 1.8],
             [7.2, 3.6, 6.1, 2.5],
             [6.5, 3.2, 5.1, 2. ],
             [6.4, 2.7, 5.3, 1.9],
             [6.8, 3. , 5.5, 2.1],
             [5.7, 2.5, 5. , 2. ],
             [5.8, 2.8, 5.1, 2.4],
             [6.4, 3.2, 5.3, 2.3],
             [6.5, 3. , 5.5, 1.8],
             [7.7, 3.8, 6.7, 2.2],
             [7.7, 2.6, 6.9, 2.3],
             [6. , 2.2, 5. , 1.5],
             [6.9, 3.2, 5.7, 2.3],
             [5.6, 2.8, 4.9, 2. ],
             [7.7, 2.8, 6.7, 2. ],
             [6.3, 2.7, 4.9, 1.8],
             [6.7, 3.3, 5.7, 2.1],
             [7.2, 3.2, 6. , 1.8],
             [6.2, 2.8, 4.8, 1.8],
             [6.1, 3. , 4.9, 1.8],
             [6.4, 2.8, 5.6, 2.1],
             [7.2, 3. , 5.8, 1.6],
             [7.4, 2.8, 6.1, 1.9],
             [7.9, 3.8, 6.4, 2. ],
             [6.4, 2.8, 5.6, 2.2],
             [6.3, 2.8, 5.1, 1.5],
             [6.1, 2.6, 5.6, 1.4],
             [7.7, 3. , 6.1, 2.3],
             [6.3, 3.4, 5.6, 2.4],
             [6.4, 3.1, 5.5, 1.8],
             [6. , 3. , 4.8, 1.8],
             [6.9, 3.1, 5.4, 2.1],
             [6.7, 3.1, 5.6, 2.4],
             [6.9, 3.1, 5.1, 2.3],
             [5.8, 2.7, 5.1, 1.9],
             [6.8, 3.2, 5.9, 2.3],
             [6.7, 3.3, 5.7, 2.5],
             [6.7, 3. , 5.2, 2.3],
             [6.3, 2.5, 5. , 1.9],
             [6.5, 3. , 5.2, 2. ],
             [6.2, 3.4, 5.4, 2.3],
             [5.9, 3. , 5.1, 1.8]]),
      'feature_names': ['sepal length (cm)',
       'sepal width (cm)',
       'petal length (cm)',
       'petal width (cm)'],
      'filename': '/usr/local/lib/python3.7/dist-packages/sklearn/datasets/data/iris.csv',
      'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
```

```
x = pd.DataFrame(df.data,columns=['sepal length (cm)',
  'sepal width (cm)',
   'sepal width (cm)',
  'petal length (cm)',
  'petal width (cm)'])
```

```
y = pd.DataFrame(df.target,columns=['target'])
```

```
df1 = pd.concat([x,y],axis=1)
```

```
df1.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

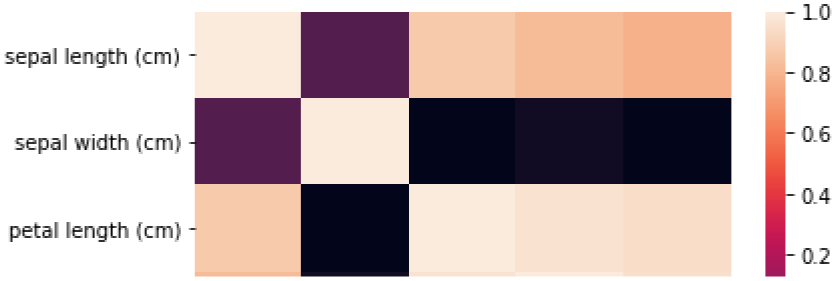
```
tc = df1.corr()
```

```
tc
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
sepal length (cm)	1.000000	-0.117570	0.871754	0.817941	0.782561
sepal width (cm)	-0.117570	1.000000	-0.428440	-0.366126	-0.426658
petal length (cm)	0.871754	-0.428440	1.000000	0.962865	0.949035

```
sns.heatmap(tc)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f764c40d590>
```



```
df1.columns
```

```
Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
       'petal width (cm)', 'target'],
      dtype='object')
```

```
x=df1[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
       'petal width (cm)']]
```

```
y = df1[['target']]
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size = 0.3)
```

```
model = RandomForestClassifier()
```

```
model.fit(x_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: DataConversionWarning: A
  """Entry point for launching an IPython kernel.
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

```
y_pred=model.predict(x_test)
```

```
metrics.accuracy_score(y_test,y_pred)
```

```
1.0
```

```
from sklearn.ensemble import AdaBoostClassifier
```

```
model = AdaBoostClassifier()
```

```
model.fit(x_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:760: DataConversionWa
  y = column_or_1d(y, warn=True)
AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None, learning_rate=1.0,
                   n_estimators=50, random_state=None)
```

```
y_pred=model.predict(x_test)
```

```
metrics.accuracy_score(y_test,y_pred)
```

```
0.9333333333333333
```

```
pip install xgboost
```

```
Requirement already satisfied: xgboost in /usr/local/lib/python3.7/dist-packages (0.90)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from xgb
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from xgb
```

```
import xgboost
```

```
from xgboost import XGBClassifier
```

```
model = XGBClassifier()
```

```
model.fit(x_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py:235: DataConversi
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py:268: DataConversi
  y = column_or_1d(y, warn=True)
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='multi:softprob', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

```
y_pred = model.predict(x_test)
```

```
metrics.accuracy_score(y_test,y_pred)
```

```
1.0
```