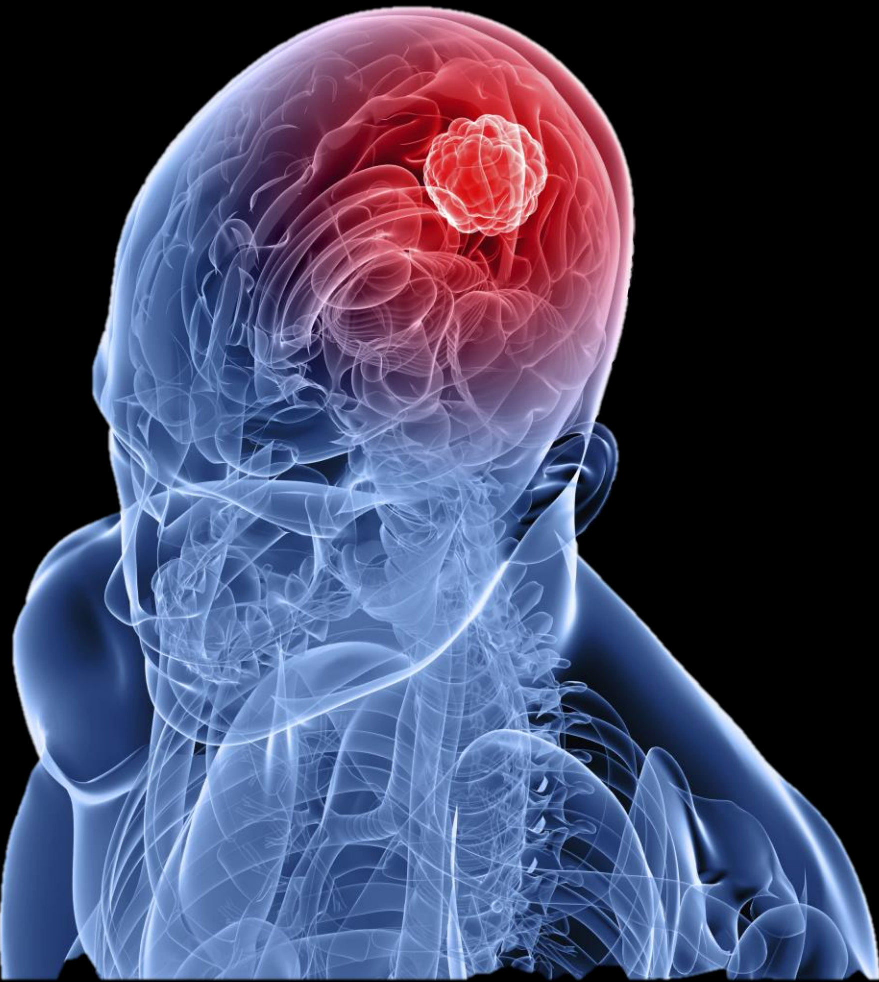
Tumour Presence Prediction using Machine Learning and Data Science

June 2018



Project by:
Vanshika
Somnath

Table of Contents

Introduction

Tumour:

- Also called malignancy, cancer, sarcoma
- Swelling of a part of body caused by abnormal growth of tissue.
- Human body contains trillions of cells, it can occur anywhere.

Machine Learning and Data Science for Tumours

- Also called malignancy, cancer, sarcoma
- Swelling of a part of body caused by abnormal growth of tissue.
- Human body contains trillions of cells, it can occur anywhere.

Elements of the Code

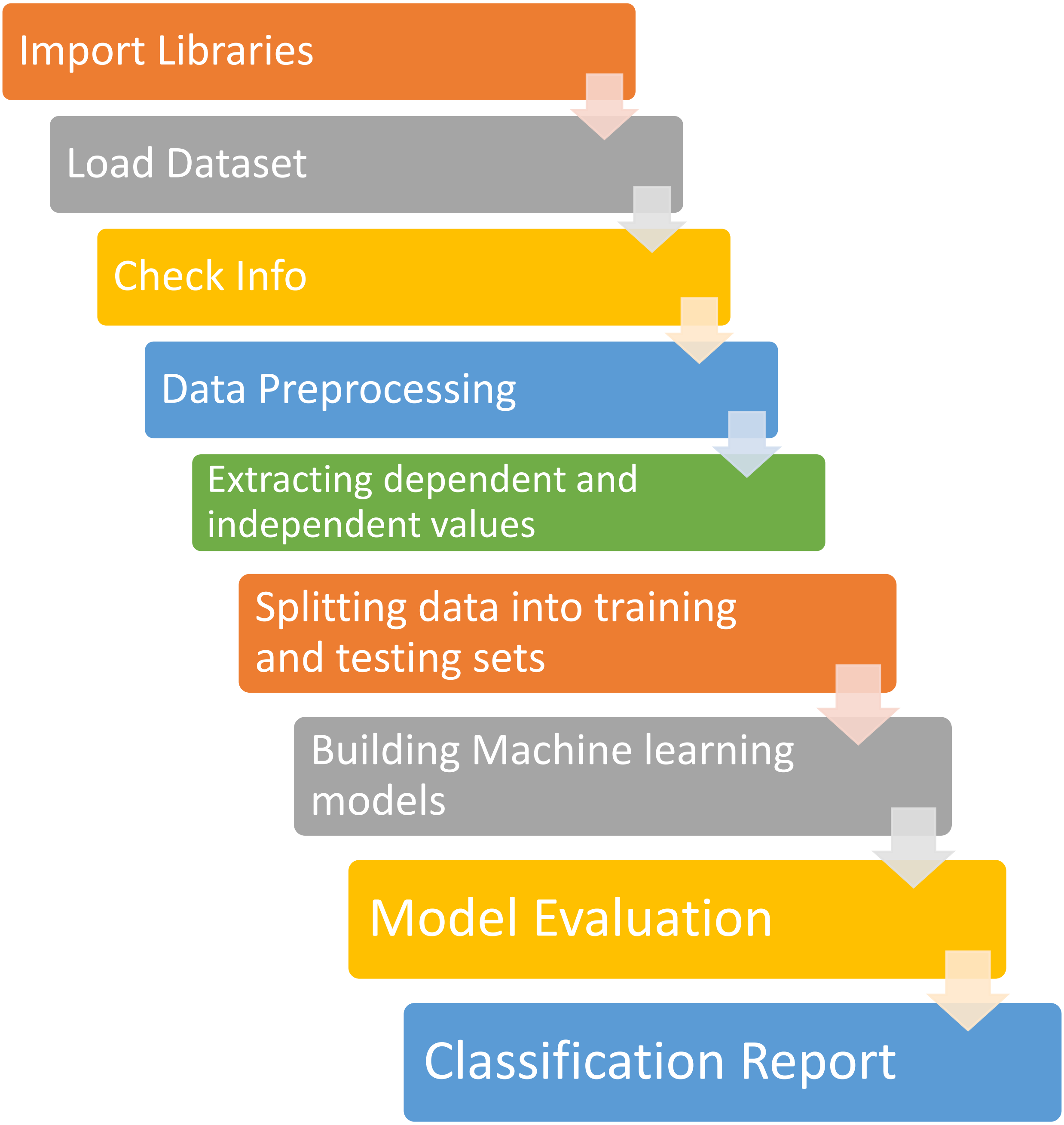
Libraries Used:

- numpy – Numerical Python
- pandas – Panel Datasets
- matplotlib
- sklearn.preprocessing
- seaborn
- sklearn.cross_validation
- sklearn.linear_model
- sklearn.neighbors
- sklearn.tree
- sklearn.svm
- sklearn.ensemble

Major Functions Used:

- dataset.head()
- dataset.info()
- dataset.keys()
- dataset.describe()
- dataset.corr()
- dataset.drop()
- plt.figure(figsize(x,y))
- plt.boxplot()
- plt.title()
- plt.show()
- x_train.shape, y_train.shape, x_test.shape, y_test.shape
- x.shape, y.shape
- sns.heatmap()
- dataset.dropna()

Process/Steps



Import Libraries

Load Dataset

Check Info

Data Preprocessing

Extracting dependent and independent values

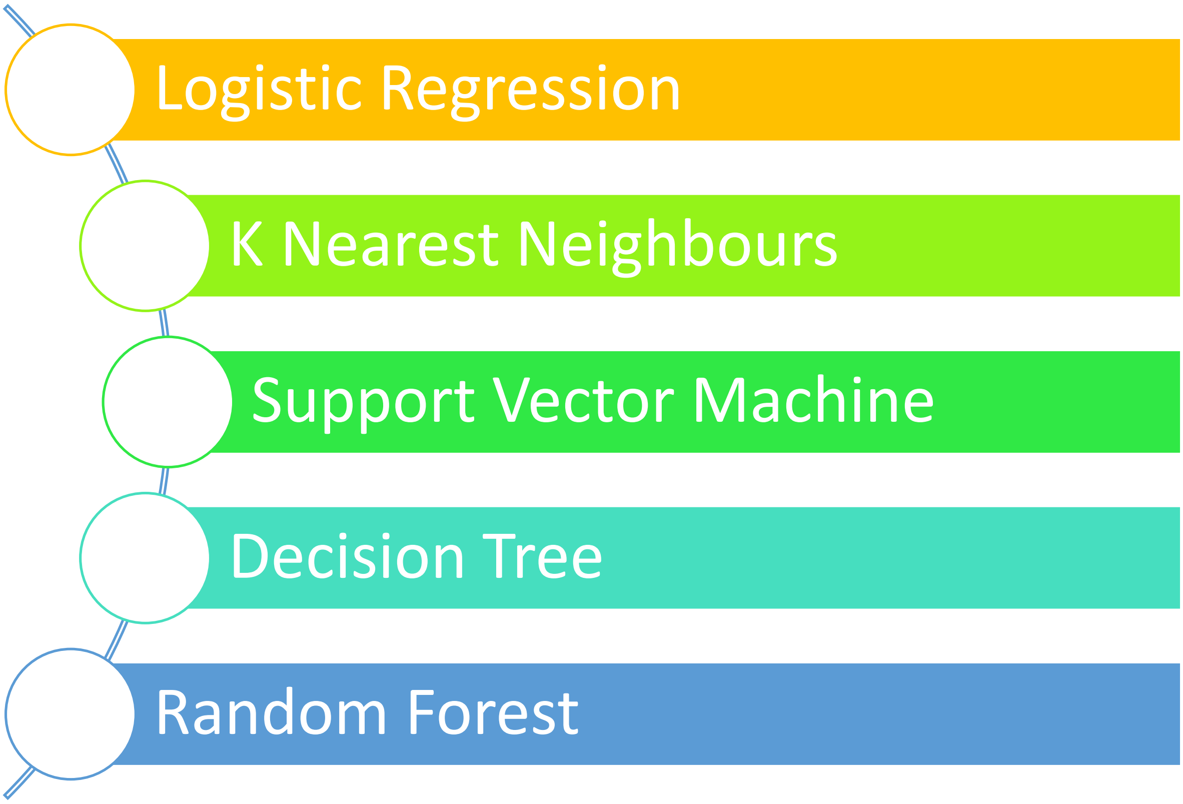
Splitting data into training and testing sets

Building Machine learning models

Model Evaluation

Classification Report

Kinds of ML Models Used:



Logistic Regression

K Nearest Neighbours

Support Vector Machine

Decision Tree

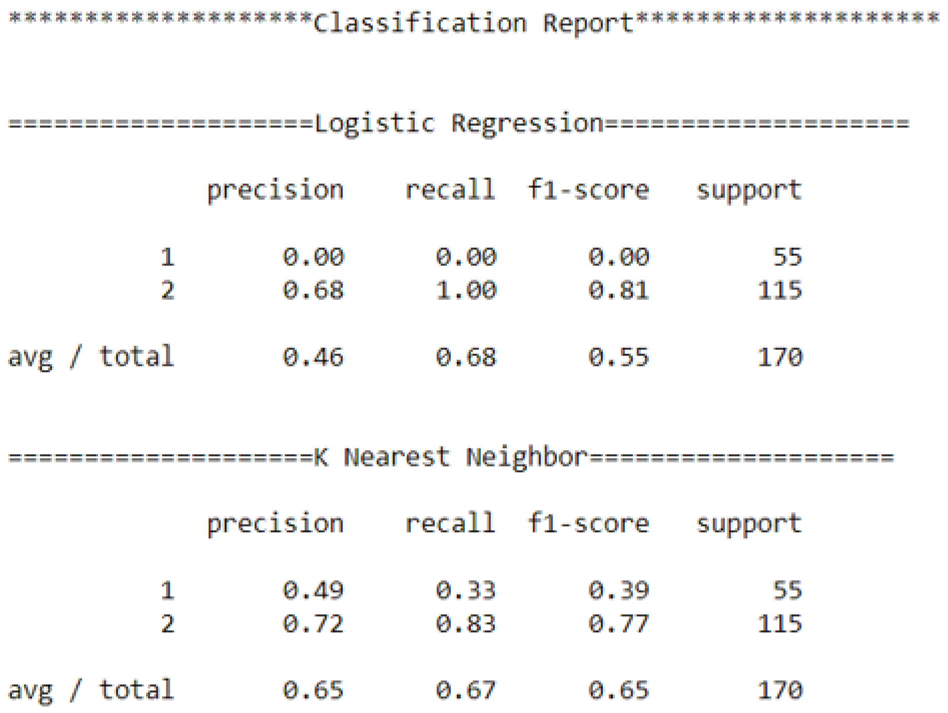
Random Forest

In the built ML model, the accuracy of the above mentioned models are as follows:

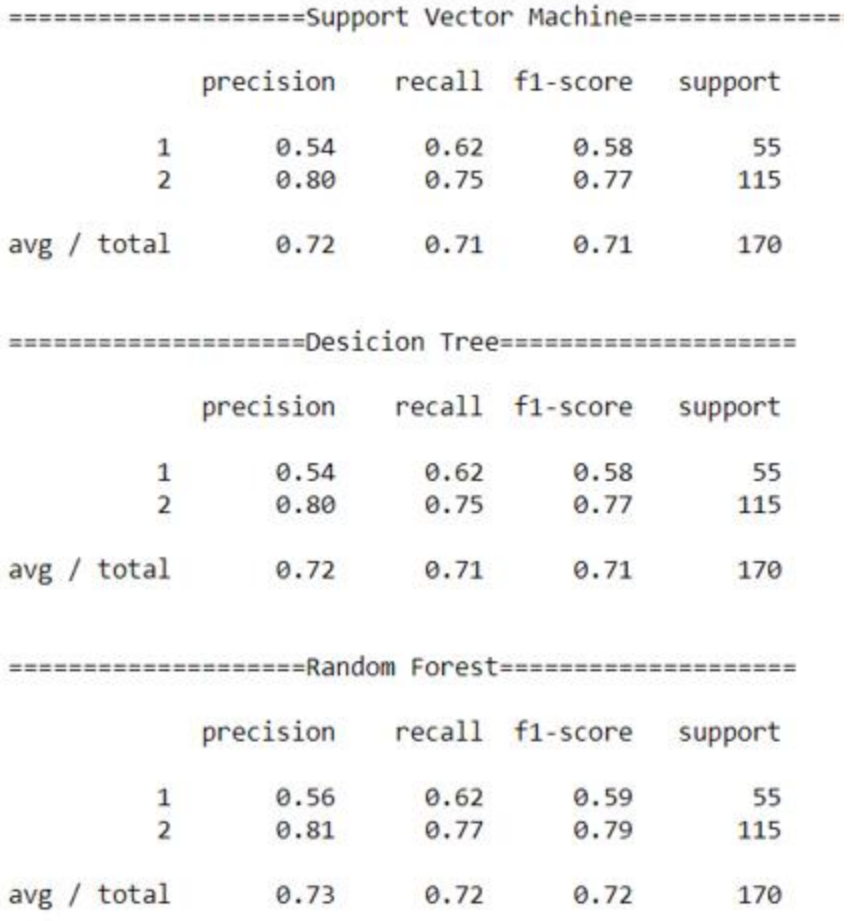
- Logistic Regression: 46%
- K Nearest Neighbours: 65%
- Support Vector Machines: 72%
- Decision Tree: 72%
- Random Forest: 73%

Hence, the highest accuracy is for Random Forest (from the classification report)

Results of the model

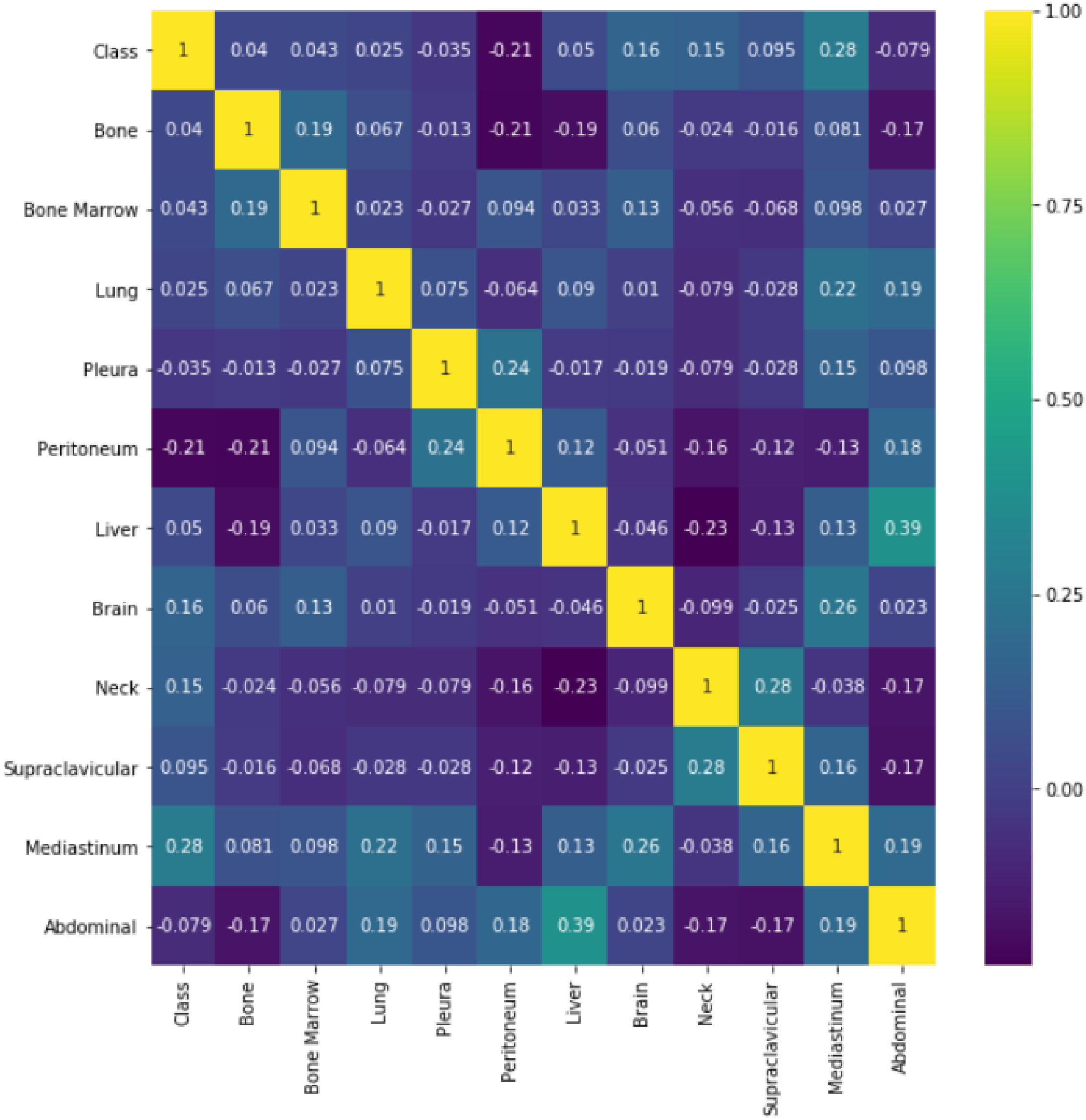


```
********************Classification Report********************


====================Logistic Regression====================

             precision    recall  f1-score   support

          1       0.00      0.00      0.00        55
          2       0.68      1.00      0.81       115

avg / total       0.46      0.68      0.55       170


====================K Nearest Neighbor====================

             precision    recall  f1-score   support

          1       0.49      0.33      0.39        55
          2       0.72      0.83      0.77       115

avg / total       0.65      0.67      0.65       170
```



```
====================Support Vector Machine==============

             precision    recall  f1-score   support

          1       0.54      0.62      0.58        55
          2       0.80      0.75      0.77       115

avg / total       0.72      0.71      0.71       170


====================Desicion Tree====================

             precision    recall  f1-score   support

          1       0.54      0.62      0.58        55
          2       0.80      0.75      0.77       115

avg / total       0.72      0.71      0.71       170


====================Random Forest====================

             precision    recall  f1-score   support

          1       0.56      0.62      0.59        55
          2       0.81      0.77      0.79       115

avg / total       0.73      0.72      0.72       170
```

Results of the model



	Class	Bone	Bone Marrow	Lung	Pleura	Peritoneum	Liver	Brain	Neck	Supraclavicular	Mediastinum	Abdominal
Class	1	0.04	0.043	0.025	-0.035	-0.21	0.05	0.16	0.15	0.095	0.28	-0.079
Bone	0.04	1	0.19	0.067	-0.013	-0.21	-0.19	0.06	-0.024	-0.016	0.081	-0.17
Bone Marrow	0.043	0.19	1	0.023	-0.027	0.094	0.033	0.13	-0.056	-0.068	0.098	0.027
Lung	0.025	0.067	0.023	1	0.075	-0.064	0.09	0.01	-0.079	-0.028	0.22	0.19
Pleura	-0.035	-0.013	-0.027	0.075	1	0.24	-0.017	-0.019	-0.079	-0.028	0.15	0.098
Peritoneum	-0.21	-0.21	0.094	-0.064	0.24	1	0.12	-0.051	-0.16	-0.12	-0.13	0.18
Liver	0.05	-0.19	0.033	0.09	-0.017	0.12	1	-0.046	-0.23	-0.13	0.13	0.39
Brain	0.16	0.06	0.13	0.01	-0.019	-0.051	-0.046	1	-0.099	-0.025	0.26	0.023
Neck	0.15	-0.024	-0.056	-0.079	-0.079	-0.16	-0.23	-0.099	1	0.28	-0.038	-0.17
Supraclavicular	0.095	-0.016	-0.068	-0.028	-0.028	-0.12	-0.13	-0.025	0.28	1	0.16	-0.17
Mediastinum	0.28	0.081	0.098	0.22	0.15	-0.13	0.13	0.26	-0.038	0.16	1	0.19
Abdominal	-0.079	-0.17	0.027	0.19	0.098	0.18	0.39	0.023	-0.17	-0.17	0.19	1

Correlation Heatmap

Scope and Research

- Deep Learning plays a vital role in the early detection of cancer. A study published by NVIDIA showed that deep learning drops error rate for breast cancer diagnoses by 85%.

- Deep learning has shown capabilities in achieving higher diagnostic accuracy results in comparison to many domain experts.

- Machine Learning alone cannot detect cancerous tumours at an early stage. Deep learning needs to be applied.

- Given dataset has been analysed and predictive machine learning model has been developed for the most likely part of body to be affected by the tumour cells.

Appendix

Python Code for the machine learning model:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

dataset =
pd.read_csv('C:/Users/Som/Desktop/Project/tumor.data
',header=None)
dataset.head()

dataset.columns = ['Class','Age','Sex','Histological
Type','Degree','Bone','Bone
Marrow','Lung','Pleura','Peritoneum','Liver','Brain'
,'Skin','Neck','Supraclavicular','Axillar','Mediasti
num','Abdominal']
dataset.head()

dataset.info()

dataset = dataset.dropna(axis=1)
dataset.head()

dataset = dataset.drop('Histological Type',1)
dataset.head()

dataset = dataset.drop('Age',1)
dataset.head()

dataset = dataset.dropna(axis=0)
dataset.head()
```

```
from sklearn.preprocessing import
LabelEncoder
le = LabelEncoder()
dataset['Class'] =
le.fit_transform(dataset['Class'])
dataset.head()

dataset.describe()

dataset.corr()

X = dataset.iloc[:,0:1].values #
independent
y = dataset.iloc[:,-1].values # dependent
X.shape, y.shape

col = dataset.keys()
col

import seaborn as sns
corr = dataset.corr()
plt.figure(figsize=(10,10))
sns.heatmap(corr,annot=True,cmap='viridis
')
plt.show()
```

```
from sklearn.cross_validation import train_test_split
x_train, x_test,y_train,y_test=
train_test_split(X,y,test_size=0.5,random_state = 0)
x_train.shape, x_test.shape, y_train.shape,y_test.shape

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

model_log = LogisticRegression (C=10.0)
model_knn = KNeighborsClassifier(n_neighbors = 3)
model_svm = SVC((C=10.0,kernel='rbf')
model_dt = DecisionTreeClassifier()
model_rf = RandomForestClassifier()
model_log.fit(x_train,y_train)
model_knn.fit(x_train,y_train)
model_svm.fit(x_train,y_train)
model_dt.fit(x_train,y_train)
model_rf.fit(x_train,y_train)

y_pred_log = model_log.predict(x_test)
y_pred_knn = model_knn.predict(x_test)
y_pred_svm = model_svm.predict(x_test)
y_pred_dt = model_dt.predict(x_test)
y_pred_rf = model_rf.predict(x_test)

from sklearn.metrics import confusion_matrix,
classification_report
cm_log = confusion_matrix(y_test,y_pred_log)
cm_knn = confusion_matrix(y_test,y_pred_knn)
cm_svm = confusion_matrix(y_test,y_pred_svm)
cm_dt = confusion_matrix(y_test,y_pred_dt)
cm_rf = confusion_matrix(y_test,y_pred_rf)
```

```
import seaborn as sns
plt.figure(figsize=(9,9))
sns.heatmap(cm_log,annot=True,cmap='summer')
plt.title('Logistic Regression')
plt.show()

plt.figure(figsize=(9,9))
sns.heatmap(cm_knn,annot=True,cmap='magma')
plt.title('K Nearest Neighbors')
plt.show()

plt.figure(figsize=(9,9))
sns.heatmap(cm_svm,annot=True,cmap='plasma')
plt.title('SVM')
plt.show()

plt.figure(figsize=(9,9))
sns.heatmap(cm_dt,annot=True,cmap='inferno')
plt.title('Decision Tree')
plt.show()

plt.figure(figsize=(9,9))
sns.heatmap(cm_rf,annot=True,cmap='PuRd')
plt.title('Random Forest')
plt.show()
```

```
print('\n'+"*"*20+ 'Classification Report'+
"*"*20+'\n')

cr_log =
classification_report(y_test,y_pred_log)
print('\n'+"="*20+ 'Logistic Regression'+
"="*20+'\n')
print(cr_log)

cr_knn =
classification_report(y_test,y_pred_knn)
print('\n'+"="*20+ 'K Nearest Neighbor'+
"="*20+'\n')
print(cr_knn)

cr_svm =
classification_report(y_test,y_pred_svm)
print('\n'+"="*20+ 'Support Vector Machine'+
"="*20+'\n')
print(cr_svm)

cr_dt =
classification_report(y_test,y_pred_dt)
print('\n'+"="*20+ 'Desicion Tree'+
"="*20+'\n')
print(cr_dt)

cr_rf=
classification_report(y_test,y_pred_rf)
print('\n'+"="*20+ 'Random Forest'+
"="*20+'\n')
print(cr_rf)
```

Sources

- ✓ UCI Machine Learning Repository
- ✓ University Medical Centre, Institute of Oncology, Yugoslavia

Tools Used

- ✓ Python
- ✓ Jupyter Notebook

Project by

Vanshika | vanshika.anumula@gmail.com
Somnath | ss.som1997@gmail.com