Glaucoma Diagnosis

Glaucoma is a gathering of eye conditions that harm the optic nerve, the soundness of which is indispensable for acceptable vision. This harm is frequently brought about by a strangely high weight in your eye.

Glaucoma is one of the main sources of visual impairment for individuals beyond 60 years old. It can happen at any age yet is progressively normal in more established grown-ups.

Numerous types of glaucoma have no admonition signs. The impact is continuous to such an extent that you may not see an adjustment in vision until the condition is at a propelled arrange.

Since vision misfortune because of glaucoma can't be recouped, it's essential to have customary eye tests that incorporate estimations of your eye pressure so a conclusion can be made in its beginning times and treated properly. On the off chance that glaucoma is perceived early, vision misfortune can be eased back or forestalled. In the event that you have the condition, you'll for the most part need treatment for a mind-blowing remainder.

In [23]:
```
#Library
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
import seaborn as sn
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
import plotly.graph_objects as go
```

In [24]:
```
#dataset
data_load = pd.read_csv('F:\Data Scientist\Gluacoma detection\GlaucomaM.csv')
```

In [25]:
```
data_load.head()
```

Out[25]:

	ag	at	as	an	ai	eag	eat	eas	ean	eai	...	tmt	tms	tmn	tmi	mr	rnf	mdic	emd	mv	Class
0	2.220	0.354	0.580	0.686	0.601	1.267	0.336	0.346	0.255	0.331	...	-0.018	-0.230	-0.510	-0.158	0.841	0.410	0.137	0.239	0.035	normal
1	2.681	0.475	0.672	0.868	0.667	2.053	0.440	0.520	0.639	0.454	...	-0.014	-0.165	-0.317	-0.192	0.924	0.256	0.252	0.329	0.022	normal
2	1.979	0.343	0.508	0.624	0.504	1.200	0.299	0.396	0.259	0.246	...	-0.097	-0.235	-0.337	-0.020	0.795	0.378	0.152	0.250	0.029	normal
3	1.747	0.269	0.476	0.525	0.476	0.612	0.147	0.017	0.044	0.405	...	-0.035	-0.449	-0.217	-0.091	0.746	0.200	0.027	0.078	0.023	normal
4	2.990	0.599	0.686	1.039	0.667	2.513	0.543	0.607	0.871	0.492	...	-0.105	0.084	-0.012	-0.054	0.977	0.193	0.297	0.354	0.034	normal

5 rows × 63 columns

In [45]:
```
data_load.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 196 entries, 0 to 195
Data columns (total 63 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   ag      196 non-null    float64
 1   at      196 non-null    float64
 2   as      196 non-null    float64
 3   an      196 non-null    float64
 4   ai      196 non-null    float64
 5   eag     196 non-null    float64
 6   eat     196 non-null    float64
 7   eas     196 non-null    float64
 8   ean     196 non-null    float64
 9   eai     196 non-null    float64
 10  abrg    196 non-null    float64
 11  abrt    196 non-null    float64
 12  abrs    196 non-null    float64
 13  abrn    196 non-null    float64
 14  abri    196 non-null    float64
 15  hic     196 non-null    float64
 16  mhcg    196 non-null    float64
 17  mhct    196 non-null    float64
 18  mhcs    196 non-null    float64
 19  mhcn    196 non-null    float64
 20  mhci    196 non-null    float64
 21  phcg    196 non-null    float64
 22  phct    196 non-null    float64
 23  phcs    196 non-null    float64
 24  phcn    196 non-null    float64
 25  phci    196 non-null    float64
 26  hvc     196 non-null    float64
 27  vbsg    196 non-null    float64
 28  vbst    196 non-null    float64
 29  vbss    196 non-null    float64
 30  vbsn    196 non-null    float64
 31  vbsi    196 non-null    float64
 32  vasg    196 non-null    float64
 33  vast    196 non-null    float64
 34  vass    196 non-null    float64
 35  vasn    196 non-null    float64
 36  vasi    196 non-null    float64
 37  vbrg    196 non-null    float64
 38  vbrt    196 non-null    float64
 39  vbrs    196 non-null    float64
 40  vbrn    196 non-null    float64
 41  vbri    196 non-null    float64
 42  varg    196 non-null    float64
 43  vart    196 non-null    float64
 44  vars    196 non-null    float64
 45  varn    196 non-null    float64
 46  vari    196 non-null    float64
 47  mdg     196 non-null    float64
 48  mdt     196 non-null    float64
 49  mds     196 non-null    float64
 50  mdn     196 non-null    float64
 51  mdi     196 non-null    float64
 52  tmg     196 non-null    float64
 53  tmt     196 non-null    float64
 54  tms     196 non-null    float64
 55  tmn     196 non-null    float64
 56  tmi     196 non-null    float64
 57  mr      196 non-null    float64
 58  rnf     196 non-null    float64
 59  mdic    196 non-null    float64
 60  emd     196 non-null    float64
 61  mv      196 non-null    float64
 62  Class   196 non-null    int32  
dtypes: float64(62), int32(1)
memory usage: 95.8 KB
```

In [26]:
```
data_load.isnull().sum()
```

Out[26]:
```
ag       0
at       0
as       0
an       0
ai       0
        ..
rnf      0
mdic     0
emd      0
mv       0
Class    0
Length: 63, dtype: int64
```

In [27]:
```
le = LabelEncoder()
```

In [28]:
```
data_load.Class = le.fit_transform(data_load.Class)
```

In [29]:
```
data_load['Class']
```

Out[29]:
```
0      1
1      1
2      1
3      1
4      1
      ..
191    0
192    0
193    0
194    0
195    0
Name: Class, Length: 196, dtype: int32
```

In [33]:
```
from sklearn.model_selection import train_test_split
```

In [34]:
```
X = data_load.drop('Class', axis='columns')
y = data_load.Class
```

In [35]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,random_state=0)
```

In [36]:
```

model = SVC(C=1.0,kernel='linear')
```

In [37]:
```
model.fit(X_train, y_train)
```

Out[37]:
```
SVC(kernel='linear')
```

In [38]:
```
model.score(X_test, y_test)
```

Out[38]:
```
0.85
```

In [39]:
```
classes1 = {
    0:'Normal',
    1:'Gulcoma',
}
```

In [40]:
```
y_predicted = model.predict(X_test)
```

In [41]:
```
y_predicted
```

Out[41]:
```
array([1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0,
       0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0])
```

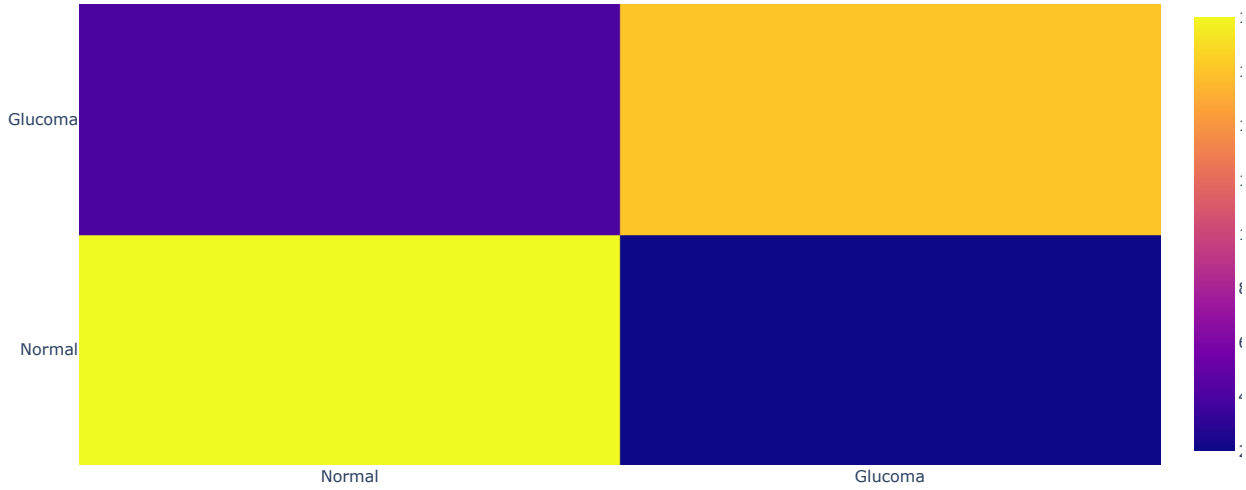
In [42]:
```
classes1[y_predicted[3]]
```

Out[42]:
```
'Normal'
```

In [43]:
```
cm = confusion_matrix(y_test, y_predicted)
cm
```

Out[43]:
```
array([[18,  2],
       [ 4, 16]], dtype=int64)
```

In [22]:
```
fig = go.Figure(data=go.Heatmap(
                    z=cm,
                    x=['Normal','Glucoma'],
                    y=['Normal','Glucoma'],
                    hoverongaps = False))
fig.show()
```



In [ ]:
```
```