Assigment # 2

Q.1

we a'realy fmaliar with linear regration in which we draw a linear line for prediction but what if our data looks like this

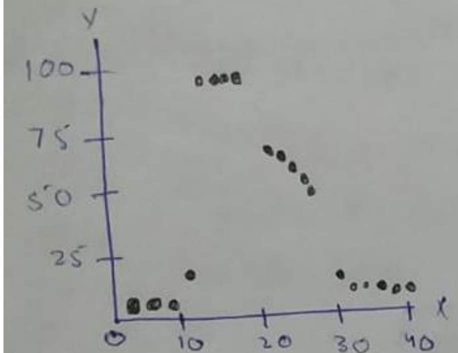


fig 1.1

Instead of

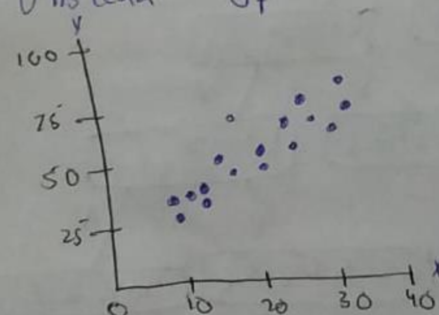
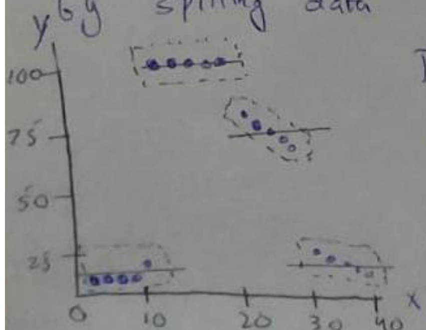


fig 1.2

In this case we must have to use a method which is not making linear line but taking desions by spliting data

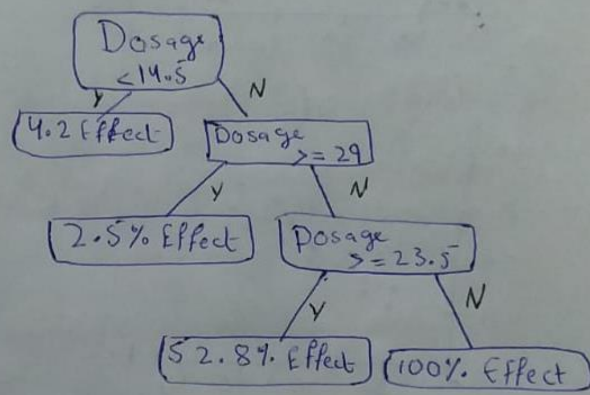


1.3

Decison tree for regration takes average value of each set as show in fig 1.3 then take dission If value of x is 12 the result would be 100

ii) Data Set

Dosage	Effect
0	0
3	0
~~5~~6	0
9	0
10	5
12	20
13	100
15	100
18	100
20	100
22	100
23	55
26	50
30	5
35	0
40	0



Q.3

Data Set = Same as question 2

To be very simple we use single feature dataset

Step # 1:

~~Average of dataset = 38.58~~

for $x_1 < 3$

Avg_1 = Avarage of other values except $X_1 = 38.8$

Sequire Residual.

$SR = (x_1 - x)^2 + (y_2 - Avg_1)^2 + (y_3 - Avg_1)^2 + (y_4 - Avg_1)^2 + \cdots$

$SR = (0-0)^2 + (0\,38.8)^2 + (0-38.8)^2 + (0-38.8)^2 + (5\,38.8)^2$
$+ (20-38.8)^2 + (100-38.8)^2 + (100-38.8)^2 + \cdots$

$SR = 27,468.5$

for $x_2 < 5$

$SR = 25,650$

for $x_3 < 7$

$SR = 20,936$

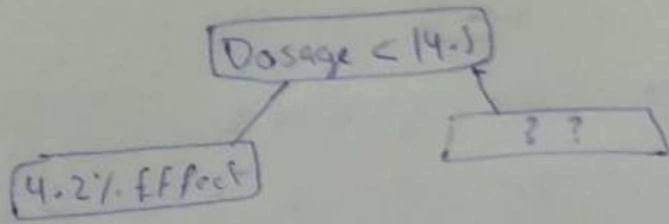
for $x_5 < 9$

$SR = 16,335$

*) In Sumerize we will take Shortest value of squire residual as a root Node and $x < 14.5$ is the smallest one

*) Then we split our data into two parts one $x < 14.5$ and 2nd one is $x >= 14.5$

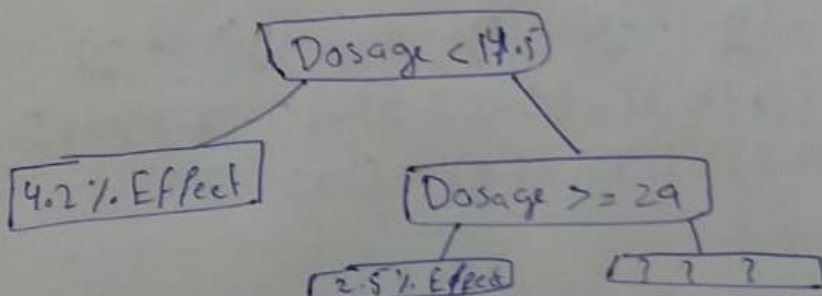
*) If we ~~hear~~. take each value as a tree node it will cause a problem of heigh Variance So that we will take 6 observation in spilited set



4.2 is the avg value of 6 observations

$0+0+0+0+\bar{5}+20 = 4.2$

Now we will take net lesser square residual which is >=29 the next Node ~~.~~ Decision node will be Dosage >=29



Now left data set consist of 9 observations and the net smallest square residual is >=23.5 So that tree will be look like this

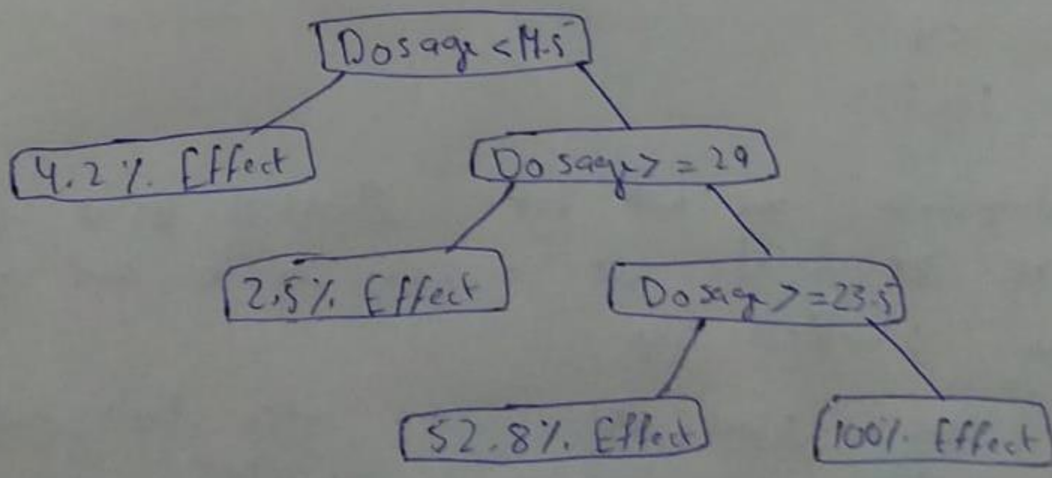
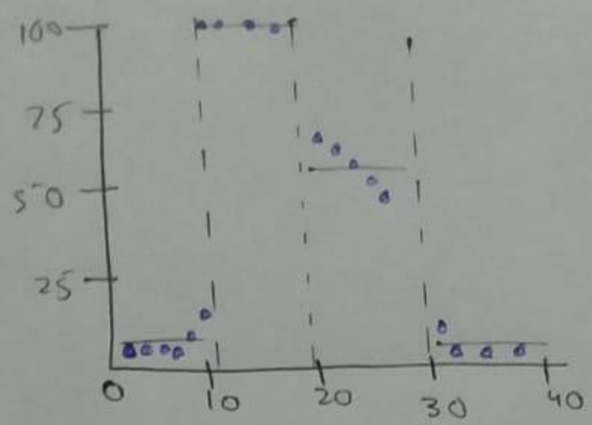


Diagram Clusters of Observations



If we have multi feature problem we take square resedual for all features and will take the lowest one as a root leaf

Q.4

Decision Tree	Sensitivity	Specificity	Accuracy	precision	AUC
IG	0.821	0.952	0.81196	0.8119	0.863
GINI	0.6752	0.918	0.6752	0.6752	0788

Gini Index Method

In [1]:
```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

In [2]:
```
dataset=pd.read_csv("Cancer_dataset.csv")
```

In [3]:
```
X=dataset.drop("Class",axis=1)
y=dataset["Class"]
```

In [4]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
```

In [14]:
```
clf = DecisionTreeClassifier(criterion="gini", max_depth=3)

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

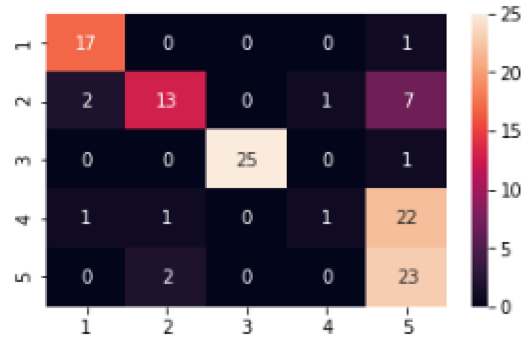
In [15]:
```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.6752136752136753
```

In [16]:
```
CM=metrics.confusion_matrix(y_test, y_pred)
```

```
In [17]: import seaborn as sn
         import pandas as pd
         import matplotlib.pyplot as plt
         df_cm = pd.DataFrame(CM, index = ['1','2','3','4','5'],columns = ['1','2','3','4','5'])
         plt.figure(figsize = (5,3))
         sn.heatmap(df_cm, annot=True)
```

```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x2ad1f2e7d30>
```




```
In [18]: C=np.array(CM)
```

```
In [19]: np.sum(C.diagonal())
```

```
Out[19]: 79
```

```
In [20]: FP = CM.sum(axis=0) - np.diag(CM)
         FN = CM.sum(axis=1) - np.diag(CM)
         TP = np.diag(CM)
         TN = CM.sum() - (FP + FN + TP)
         FP=sum(FP)
         FN=sum(FN)
         TP=sum(TP)
         TN=sum(TN)
```

```
In [21]: # Sensitivity, hit rate, recall, or true positive rate
         TPR = TP/(TP+FN)
         print('Sensitivity',TPR)
         # Specificity or true negative rate
         TNR = TN/(TN+FP)
         print('\nSpecificity',TNR)
         # Precision or positive predictive value
         PPV = TP/(TP+FP)
         print("\nPrecision",PPV)
```

```
Sensitivity 0.6752136752136753

Specificity 0.9188034188034188

Precision 0.6752136752136753
```

```
In [22]: fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred, pos_label=5)
         metrics.auc(fpr, tpr)
```

```
Out[22]: 0.7884782608695652
```

Information Gain

In [1]:
```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

In [2]:
```
dataset=pd.read_csv("Cancer_dataset.csv")
```

In [7]:
```
X=dataset.drop("Class",axis=1)
y=dataset["Class"]
```

In [9]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
```

In [10]:
```
clf = DecisionTreeClassifier()

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

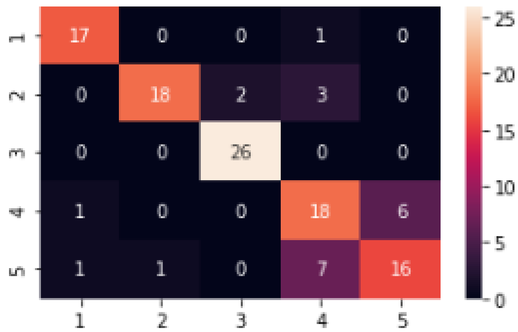
In [11]:
```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.811965811965812
```

In [15]:
```
CM=metrics.confusion_matrix(y_test, y_pred)
```

In [16]:
```python
import seaborn as sn
import pandas as pd
import matplotlib.pyplot as plt
df_cm = pd.DataFrame(CM, index = ['1','2','3','4','5'],columns = ['1','2','3','4','5'])
plt.figure(figsize = (5,3))
sn.heatmap(df_cm, annot=True)
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x22de3c8d978>



In [17]:
```python
C=np.array(CM)
```

In [19]:
```python
np.sum(C.diagonal())
```

Out[19]: 95

In [37]:
```python
FP = CM.sum(axis=0) - np.diag(CM)
FN = CM.sum(axis=1) - np.diag(CM)
TP = np.diag(CM)
TN = CM.sum() - (FP + FN + TP)
FP=sum(FP)
FN=sum(FN)
TP=sum(TP)
```

In [38]:
```python
# Sensitivity, hit rate, recall, or true positive rate
TPR = TP/(TP+FN)
print('Sensitivity',TPR)
# Specificity or true negative rate
TNR = TN/(TN+FP)
print('\nSpecificity',TNR)
# Precision or positive predictive value
PPV = TP/(TP+FP)
print("\nPrecision",PPV)
```

```
Sensitivity 0.811965811965812

Specificity 0.9529914529914529

Precision 0.811965811965812
```

In [39]:
```python
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred, pos_label=5)
metrics.auc(fpr, tpr)
```

Out[39]: 0.8630434782608696

CODE LINK: https://github.com/huxe/Machine-learning/tree/master/assig2