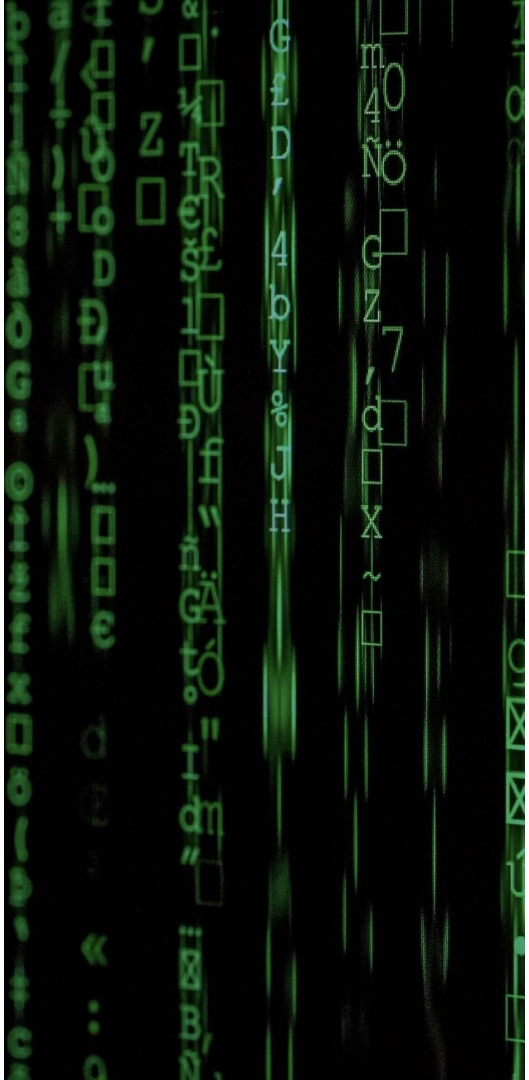


Breast Cancer Tumor Stage Prediction

Presented by:

Yun Lin,

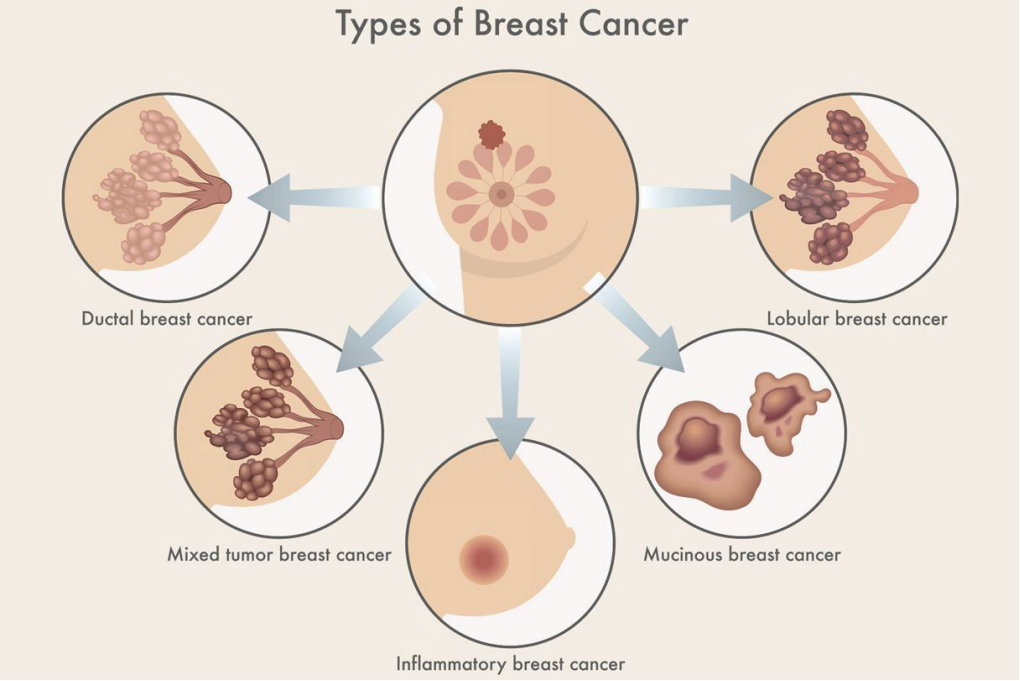
Yupan Wang,

Sangyu Baek,

Hongkai Lou

Breast Cancer

- The second most prevalent kind of cancer among women in the US
- About 264,000 women and 2,400 men are given the diagnosis of breast cancer each year in the US
- Can be caused by hormonal factors or genetic factors



Staging

- The process to determine **how far cancer cells have spread.**
- Ranges from **stage 0** to **stage 4.**
- Helpful for surgeon if stage can be classified **before operation.**
- In the process of staging, **7 key pieces of information** are used:

  The size of the tumor
  The spread to nearby lymph nodes
  The metastasis to distant sites
  Estrogen Receptor (ER) status
  Progesterone Receptor (PR) status
  HER2 status
  Grade of the cancer

Dataset Cleaning

Clinical profiles of **2,509** breast cancer patients with **34** columns of data type
Cleaning the data:

- Remove redundant columns that
  a. Cannot be **Quantifiable** for model
  b. Informations **that cannot be acquired** initially (Survival Month)

Some Removed Columns

- Vital Status
- Sex (All female)
- Relapse Status
- Survival Month/Status
- Radio Therapy
- Type of Surgery
- Hormone Therapy
- Integrative Cluster

Dataset Imputation

- Using the method of **imputation** to deal with missing value
- Numeric Variables: Mean
  a. Age
  b. Nottingham Index
- Categorical Variables: Highest Frequency
  a. Cancer Subtype
  b. Tumor History
- Variables that have too much missing values
- We have 21 Variables left at last. Only three are continuous numerical variables.

Visualization of Stage

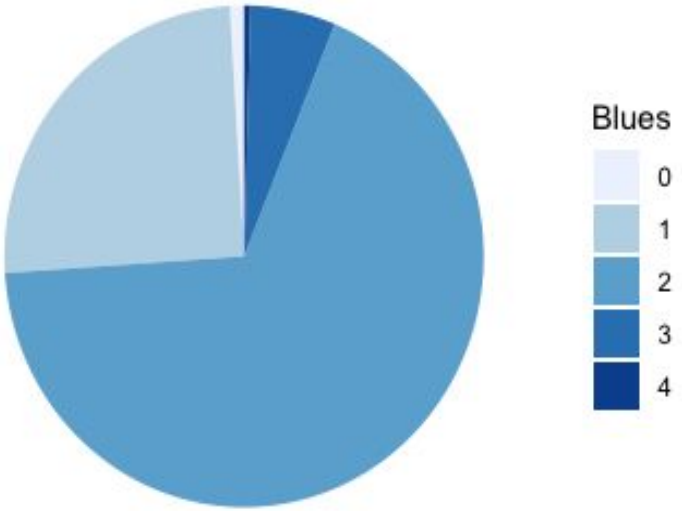
Stage 0 : 24

Stage 1: 630

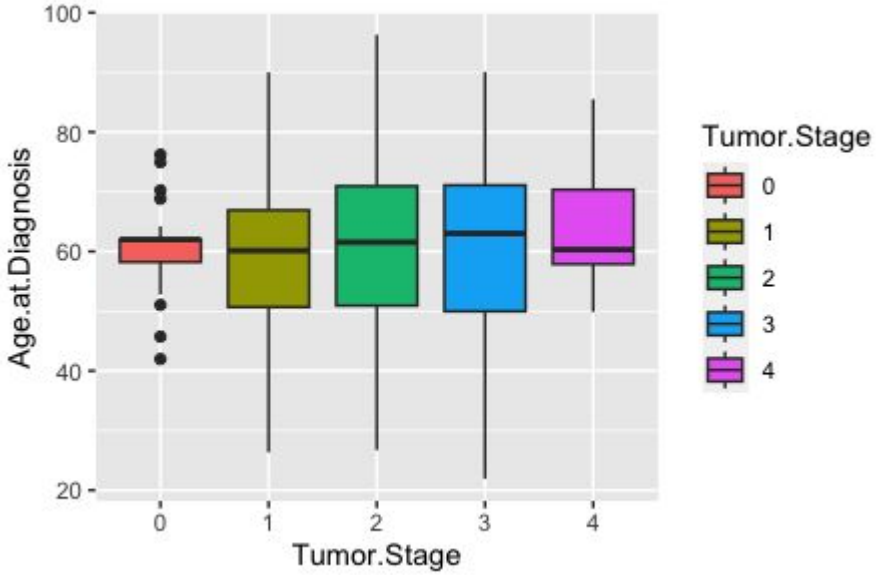
Stage 2: 1700

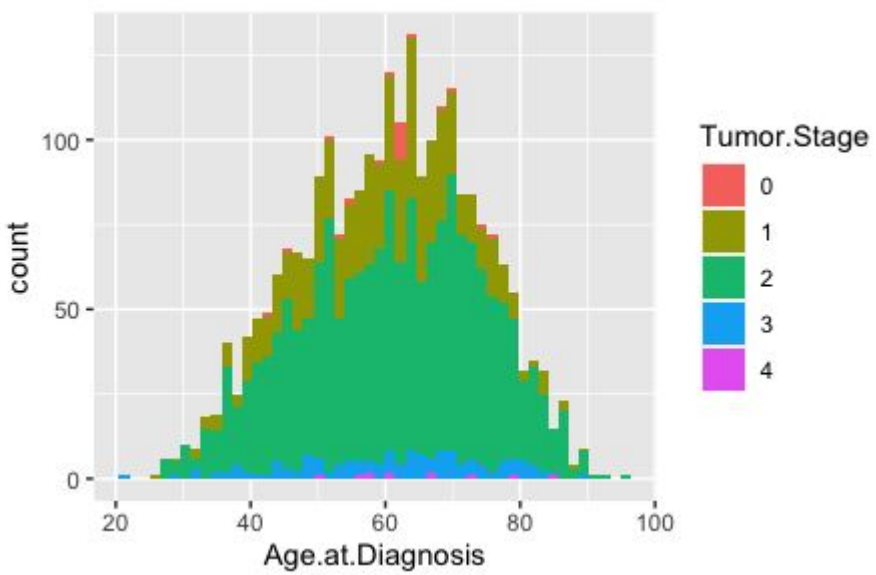
Stage 3: 144

Stage 4: 11

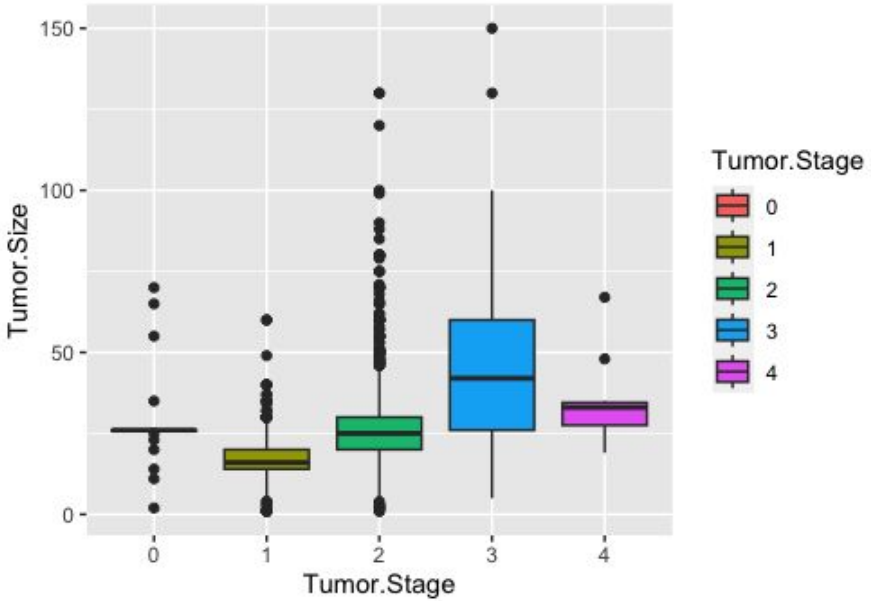


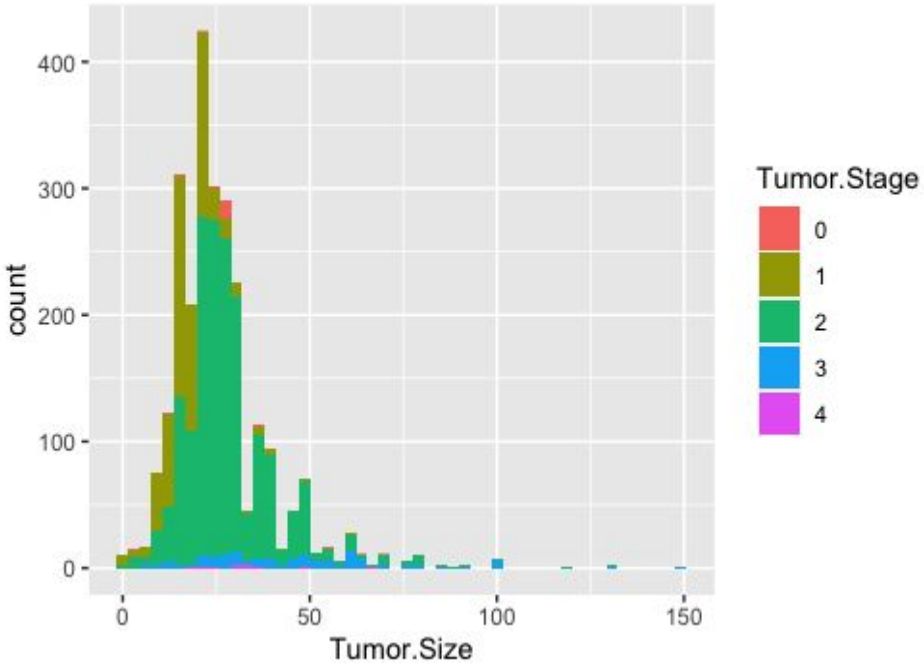
Visualization of Age and Tumor Stage



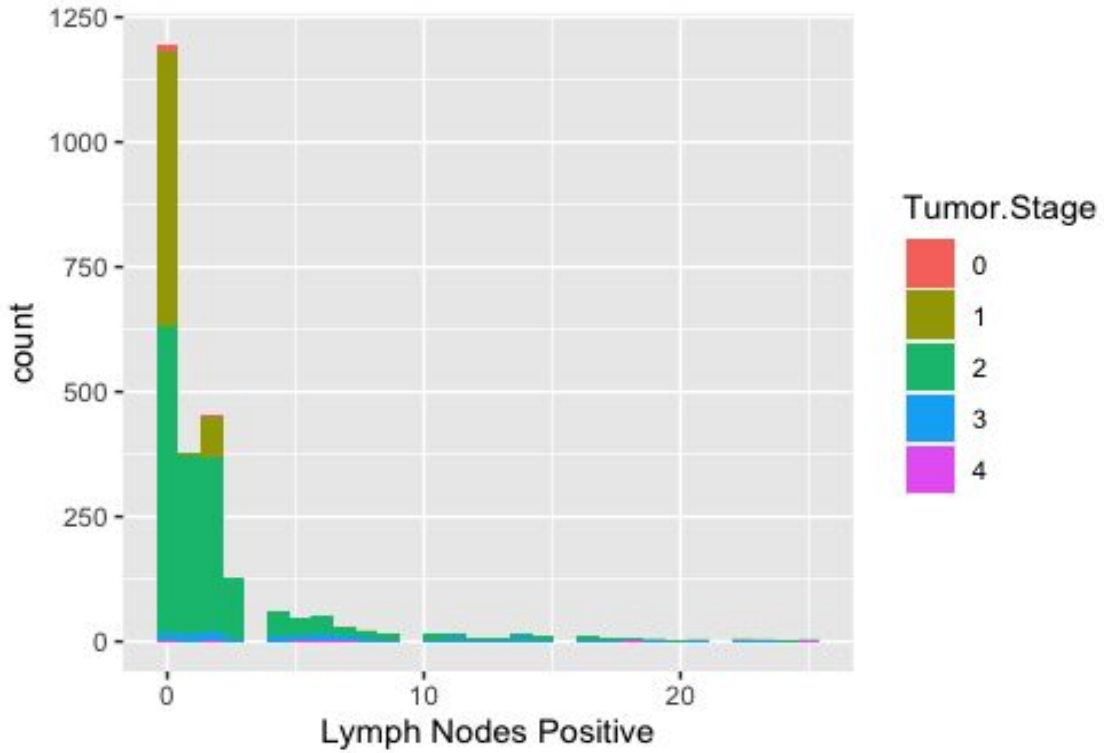


Visualization of Tumor Size and Stage

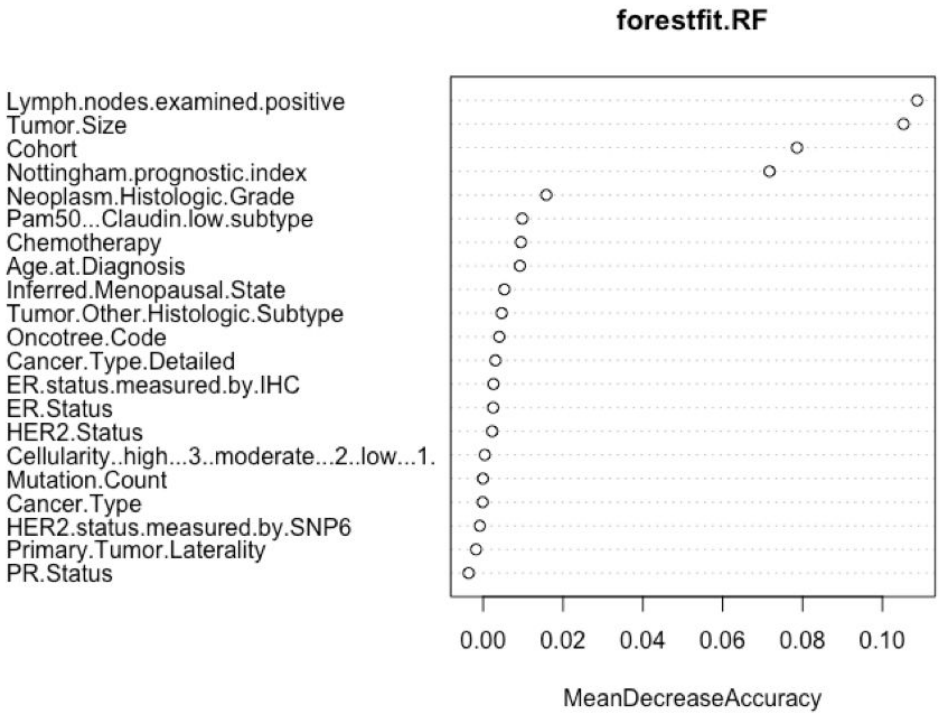




Visualization of Lymph Nodes and Tumor Stage



Some Feature Selection through Random Forest



Encode

Encoding the categorical predicators -> 0, 1, 2 . . .

```
data.head()
```

	Age.at.Diagnosis	Cancer.Type	Cancer.Type.Detailed	Cellularity..high...3..moderate...2..low...1.	Chemotherapy	Pam50...Claudin.low.subtype	Cohort	ER.status.me
0	75.65	Breast Cancer	Breast Invasive Ductal Carcinoma	Moderate	No	claudin-low	1	
1	43.19	Breast Cancer	Breast Invasive Ductal Carcinoma	High	No	LumA	1	
2	48.87	Breast Cancer	Breast Invasive Ductal Carcinoma	High	yes	LumB	1	
3	47.68	Breast Cancer	Breast Mixed Ductal and Lobular Carcinoma	Moderate	yes	LumB	1	
4	76.97	Breast Cancer	Breast Mixed Ductal and Lobular Carcinoma	High	yes	LumB	1	

5 rows × 22 columns

```
data.head()
```

	Age.at.Diagnosis	Cancer.Type	Cancer.Type.Detailed	Cellularity..high...3..moderate...2..low...1.	Chemotherapy	Pam50...Claudin.low.subtype	Cohort	ER.status.me
0	75.65	0	2	2	0	6	1	
1	43.19	0	2	0	0	2	1	
2	48.87	0	2	0	1	3	1	
3	47.68	0	5	2	1	3	1	
4	76.97	0	5	0	1	3	1	

5 rows × 22 columns

```
<AxesSubplot:>
```

Model	CV training score
Linear regression	76%
Linear regression w Lasso	71.3%
Linear regression w Ridge	73.1 %
Neural Network w 1 hidden layer	77.09%
Neural Network w 2 hidden layer	76.69%
Random Forest	87.48%
SVM	74.7%

Since the LR model output is numerical, and our desire output is classes.

So we set the different cut-off line for these numeric output as category output.

```
print('Correct Training Classification Rate is:',round(sum(y_pred==y_train)/len(y_train),3) *100 ,'%')
print('MSE Training for LR:', sum((y_pred-y_train)**2)/len(y_train))
```

```
Correct Training Classification Rate is: 0.0 %
MSE Training for LR: 0.2171131057406148
```

```
for i in range(len(y_pred)):
    if y_pred[i] < 0.5:
        y_pred[i] = 0
    elif y_pred[i] < 1.5:
        y_pred[i] = 1
    elif y_pred[i] < 2.5:
        y_pred[i] = 2
    elif y_pred[i] < 3.5:
        y_pred[i] = 3
    else:
        y_pred[i] = 4
```

```
print('Correct Training Classification Rate is:',round(sum(y_pred==y_train)/len(y_train),3) *100 ,'%')
print('MSE Training for LR:', sum((y_pred-y_train)**2)/len(y_train))
```

```
Correct Training Classification Rate is: 74.3 %
MSE Training for LR: 0.2909815645241654
```

Lasso

In [314]:
```
reg_lasso = linear_model.Lasso(alpha=0.1)
reg_lasso.fit(X_train, y_train)
y_pred = reg_lasso.predict(X_test)
```

In [315]:
```
for i in range(len(y_pred)):
    if y_pred[i] < 0.5:
        y_pred[i] = 0
    elif y_pred[i] < 1.5:
        y_pred[i] = 1
    elif y_pred[i] < 2.5:
        y_pred[i] = 2
    elif y_pred[i] < 3.5:
        y_pred[i] = 3
    else:
        y_pred[i] = 4
```

In [316]:
```
print('Classification Rate for Lasso is:',round(sum(y_test==y_pred)/len(y_pred),3) *100 ,'%')
```

```
Classification Rate for Lasso is: 71.3 %
```

Ridge ¶

In [317]:
```
#from sklearn import linear_model
reg_ridge = linear_model.Ridge(alpha=.5)
reg_ridge.fit(X_train, y_train)
y_pred = reg_ridge.predict(X_test)
```

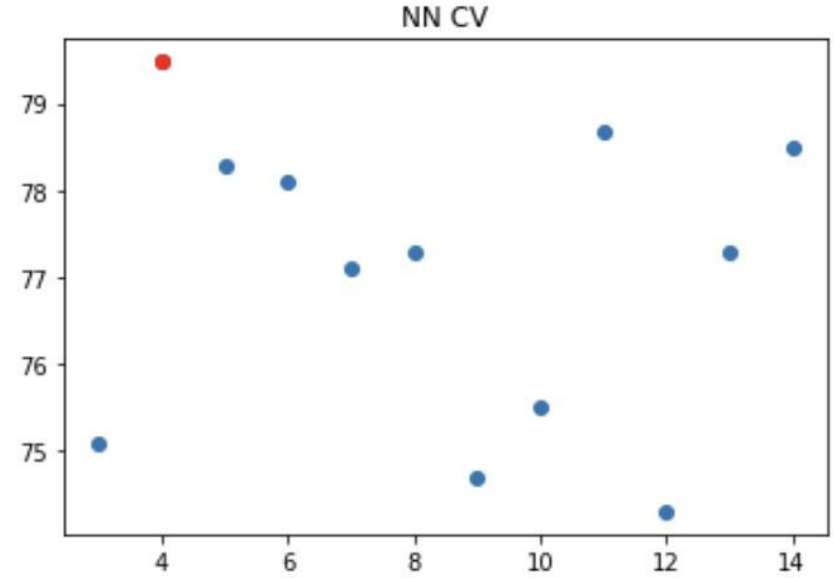
In [318]:
```
for i in range(len(y_pred)):
    if y_pred[i] < 1:
        y_pred[i] = 0
    elif y_pred[i] < 1.5:
        y_pred[i] = 1
    elif y_pred[i] < 2.5:
        y_pred[i] = 2
    elif y_pred[i] < 3.3:
        y_pred[i] = 3
    else:
        y_pred[i] = 4
```

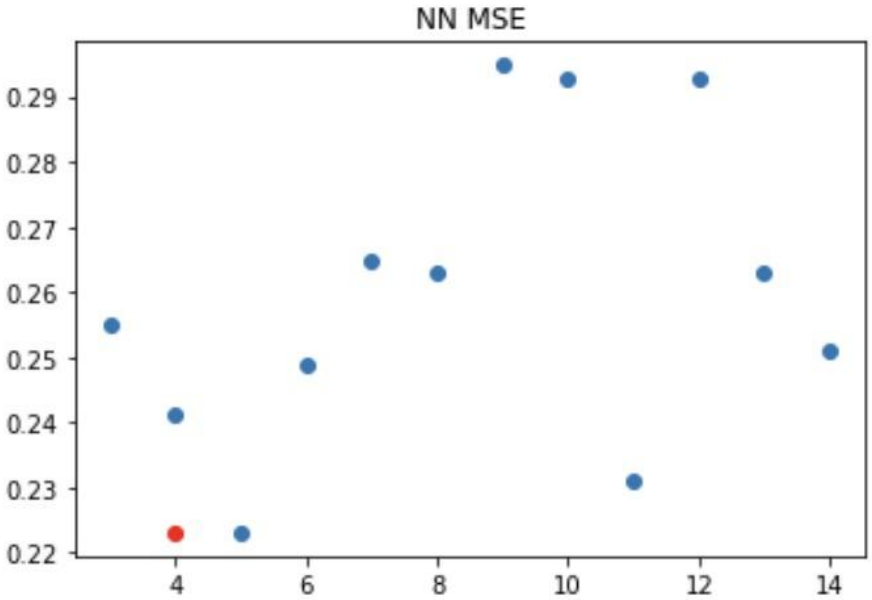
In [319]:
```
print('Classification Rate for Ridge is:',round(sum(y_test==y_pred)/len(y_pred),3) *100 ,'%')
```

```
Classification Rate for Ridge is: 73.1 %
```

Neural network with one hidden Layer

```
Best CV scroe for NN at 79.48 %
```





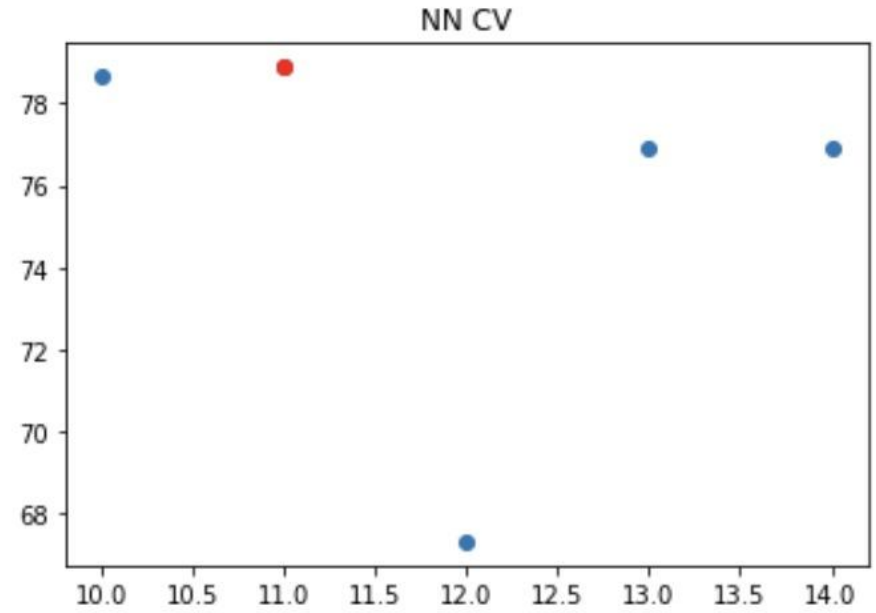
2. The most appropriate number of hidden neurons is

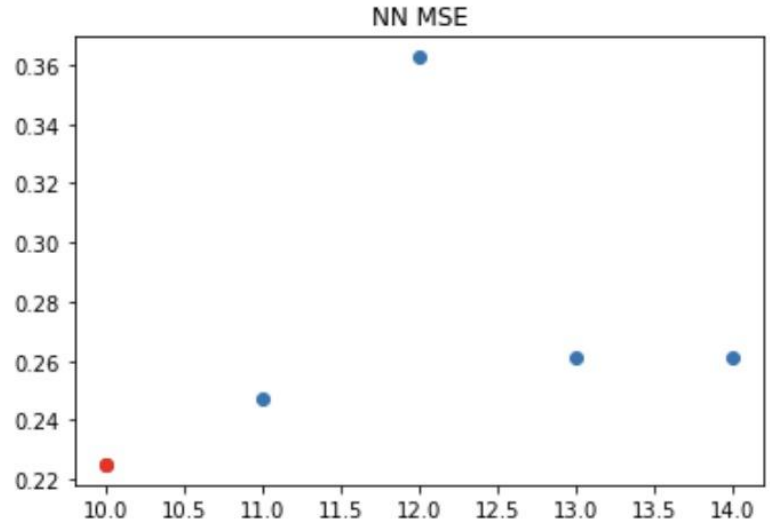
sqrt(input layer nodes * output layer nodes)

NN with two hidden Layer

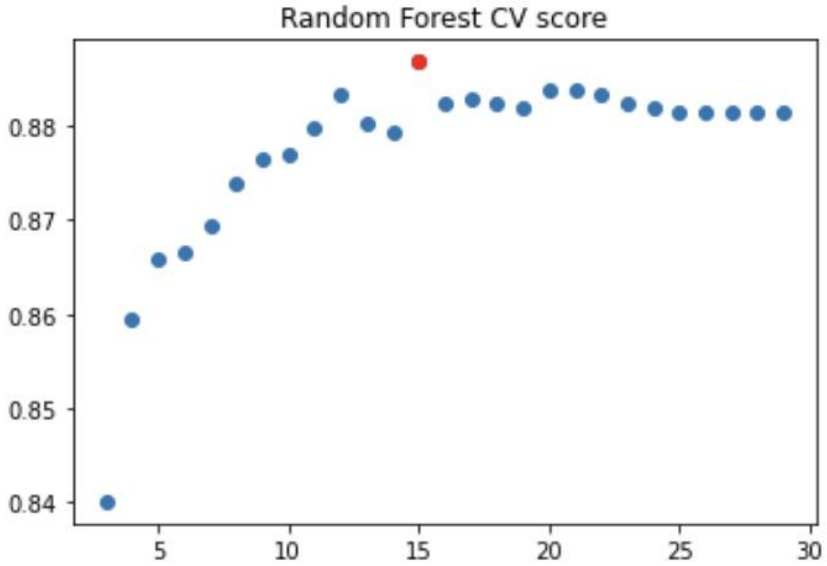
```
,hidden_layer_sizes=(i,int(i**0.5),)
```

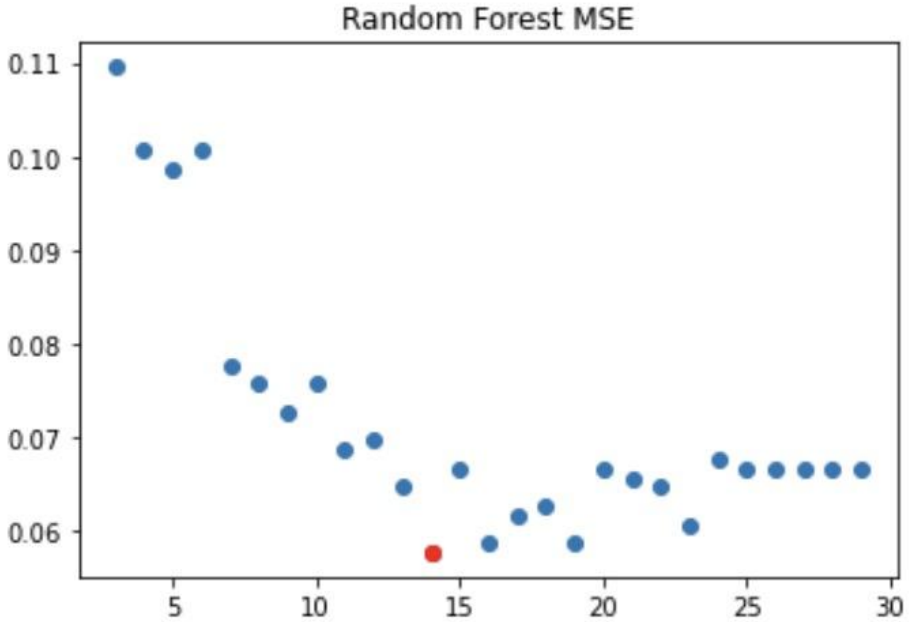
```
Best CV scroe for NN at 78.88 %
```





Random Forest





Test Scores of Random Forest method with different depths

```
Correct Test Classification Rate is: 85.5 %
MSE: 0.15139442231075698
5 110 379 8 0
Correct Test Classification Rate is: 86.1 %
MSE: 0.1454183266932271
5 111 374 12 0
Correct Test Classification Rate is: 86.1 %
MSE: 0.15139442231075698
5 116 371 10 0
Correct Test Classification Rate is: 86.3 %
MSE: 0.13745019920318724
5 117 366 14 0
Correct Test Classification Rate is: 86.7 %
MSE: 0.1394422310756972
5 117 365 15 0
Correct Test Classification Rate is: 87.6 %
MSE: 0.1294820717131474
5 123 358 16 0
Correct Test Classification Rate is: 88.4 %
MSE: 0.11553784860557768
5 119 361 17 0
```

```
print(sum(y_test==0),sum(y_test==1
```

```
5 125 338 33 1
```

Conclusion

- Regression Models failed to give high score for accuracy.
- Random Forest gives the highest score. However, it fails to identify any stage 4 tumors from the given dataset. Also, there is a high chance of overfitting with the given depth considering the number of input features we are using.
- Neural Network, although it does not yield the highest accuracy score, yields the most relevant result.
- For further improvement, we would like to try backpropagation for the final submission.

Thanks for listening ~