1 IN 8 WOMEN
in the United States will develop
breast cancer in her lifetime.

Breast cancer analysis

Prepared for: All

Prepared by: Mid Eum Lee

April 4, 2020

Summary

Problem statement

Breast cancer is the most common cancer type in women. It’s important to accurately diagnose malignancy at an early stage for improving survival. Even experienced doctors often fail to predict cancer and show 79% accuracy in diagnosis. I will use machine learning techniques to improve the method of malignancy prediction so that we can increase the accuracy in cancer prediction.

Description of the dataset

**Dataset Information : Data obtained from UCI Machine Learning Repository:https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Attribute Information:

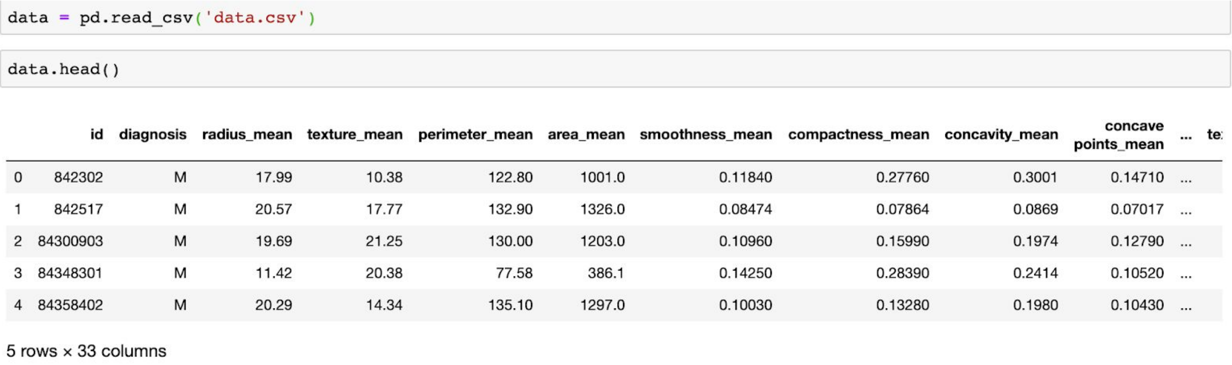
* 1) ID number
* 2) Diagnosis (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

* a) radius (mean of distances from center to points on the perimeter)
* b) texture (standard deviation of gray-scale values)
* c) perimeter
* d) area
* e) smoothness (local variation in radius lengths)
* f) compactness (perimeter^2 / area - 1.0)
* g) concavity (severity of concave portions of the contour)
* h) concave points (number of concave portions of the contour)
* i) symmetry
* j) fractal dimension ("coastline approximation" - 1)

Exploratory data analysis (data story and inferential statistics)

The initial 5 rows and 33 columns of the dataset are loaded.

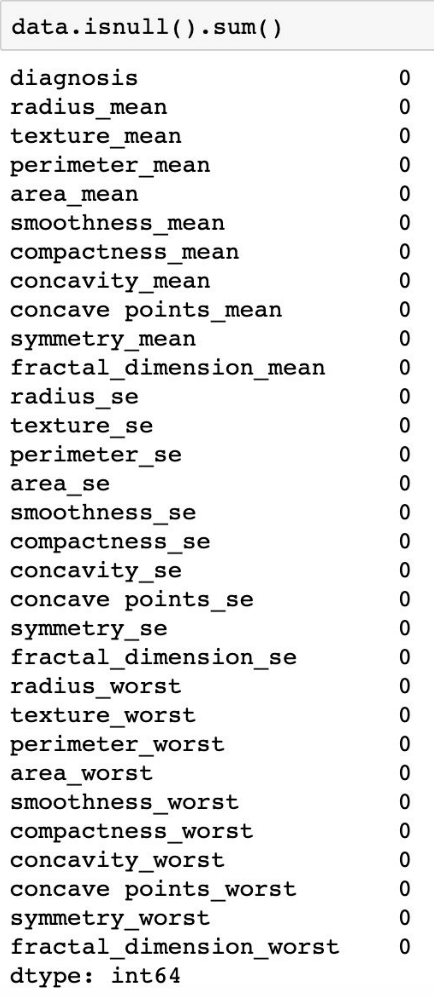


```
data = pd.read_csv('data.csv')
```

```
data.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	te
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	

5 rows × 33 columns

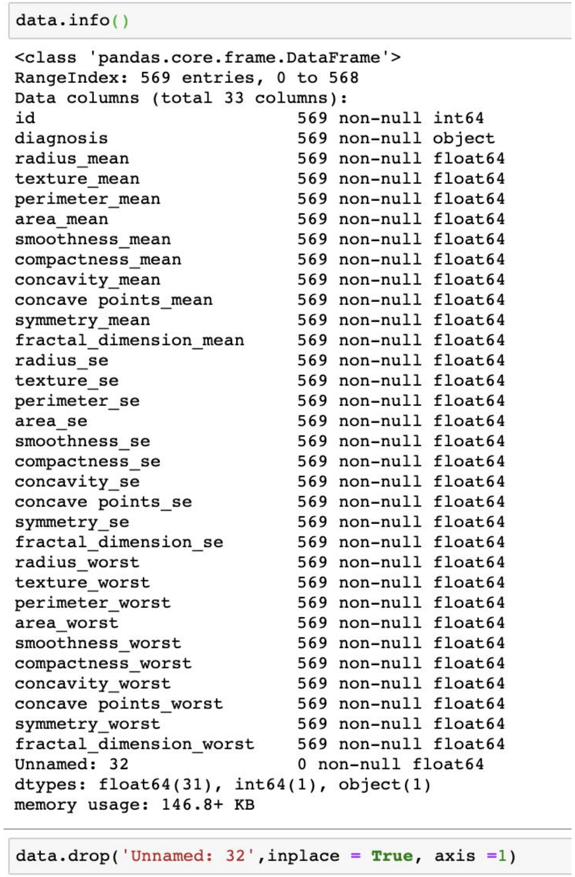


```
data.isnull().sum()
```

```
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
concavity_worst            0
concave points_worst       0
symmetry_worst             0
fractal_dimension_worst    0
dtype: int64
```

The dataset was examined for the null values. There are no null values.

The 'unnamed: 32' column contains no entries so this column was removed in the dataframe.



```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
id                         569 non-null int64
diagnosis                  569 non-null object
radius_mean                569 non-null float64
texture_mean               569 non-null float64
perimeter_mean             569 non-null float64
area_mean                  569 non-null float64
smoothness_mean            569 non-null float64
compactness_mean           569 non-null float64
concavity_mean             569 non-null float64
concave points_mean        569 non-null float64
symmetry_mean              569 non-null float64
fractal_dimension_mean     569 non-null float64
radius_se                  569 non-null float64
texture_se                 569 non-null float64
perimeter_se               569 non-null float64
area_se                    569 non-null float64
smoothness_se              569 non-null float64
compactness_se             569 non-null float64
concavity_se               569 non-null float64
concave points_se          569 non-null float64
symmetry_se                569 non-null float64
fractal_dimension_se       569 non-null float64
radius_worst               569 non-null float64
texture_worst              569 non-null float64
perimeter_worst            569 non-null float64
area_worst                 569 non-null float64
smoothness_worst           569 non-null float64
compactness_worst          569 non-null float64
concavity_worst            569 non-null float64
concave points_worst       569 non-null float64
symmetry_worst             569 non-null float64
fractal_dimension_worst    569 non-null float64
Unnamed: 32                0 non-null float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

```
data.drop('Unnamed: 32',inplace = True, axis =1)
```

Summary of findings:

1. Features like fractal_dimension_mean, texture_se, smoothness_se, symmetry_se, fractal_dimension_se can't differentiate malignant versus benign.

2. There are many outliers in texture of mass compared to the mean radius of mass. The size of mass (radius_mean) will be a better criteria for malignancy diagnosis compared to the texture of mass.

3.There is not much difference in smoothness of mass between groups. The compactness is higher in the malignancy group.

4. There are many outliers in concavity. Concave point is higher in the malignancy group.

5. The perimeter and the area of mass are higher in the malignancy group.

6. area_mean feature is highly correlated with radius_mean, area_worst and perimeter_mean.

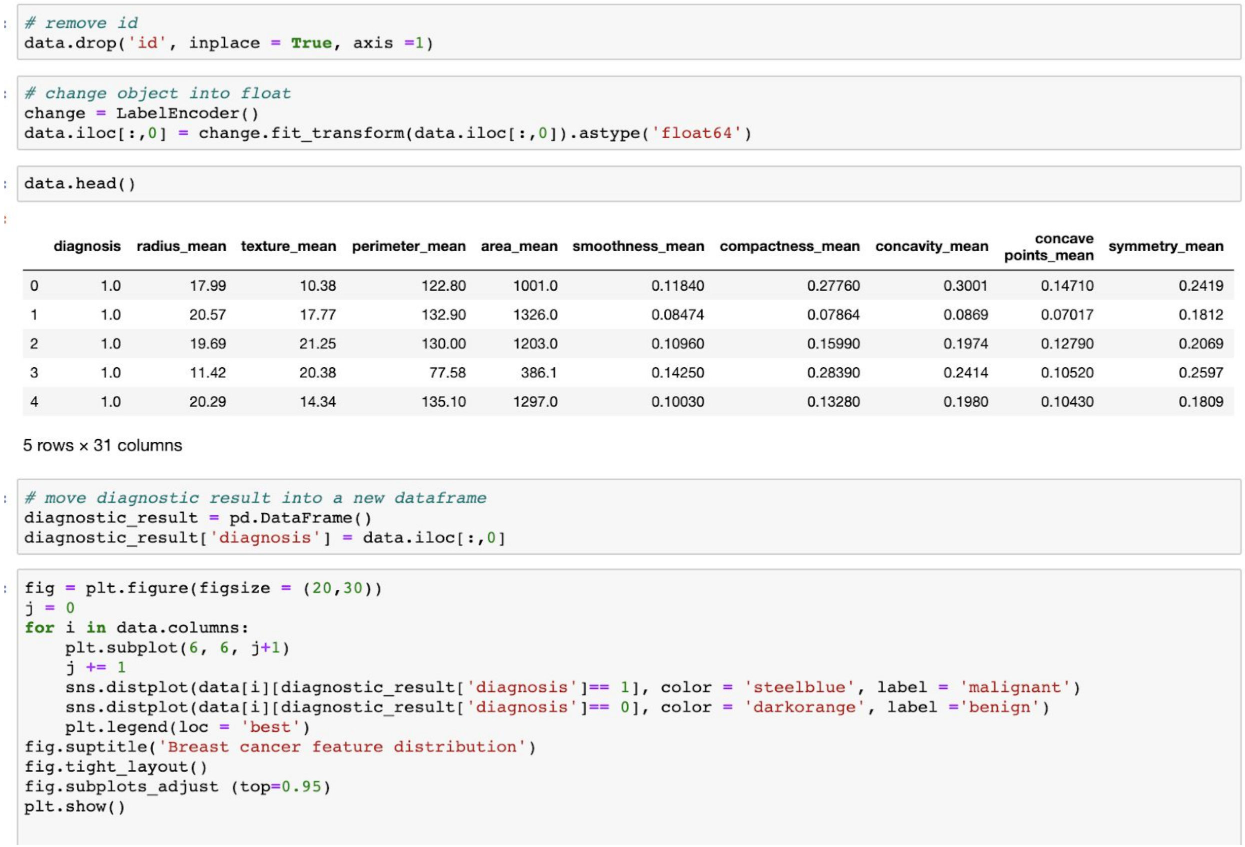
7. area_mean feature is not highly correlated with 'texture_worst','texture_mean','compactness_worst','concave points_worst' and 'fractal_dimension_worst'.

Visuals and statistics to support findings

1. Features like fractal_dimension_mean, texture_se, smoothness_se, symmetry_se, fractal_dimension_se can't differentiate malignant versus benign.

First, the column ‘ID’ was removed since this information is not needed for the analysis. The object variables M or B in the diagnosis column are converted into float64 format using LableEncoder().

The features were plotted against diagnosis using subplots.



```
# remove id
data.drop('id', inplace = True, axis =1)
```

```
# change object into float
change = LabelEncoder()
data.iloc[:,0] = change.fit_transform(data.iloc[:,0]).astype('float64')
```

```
data.head()
```

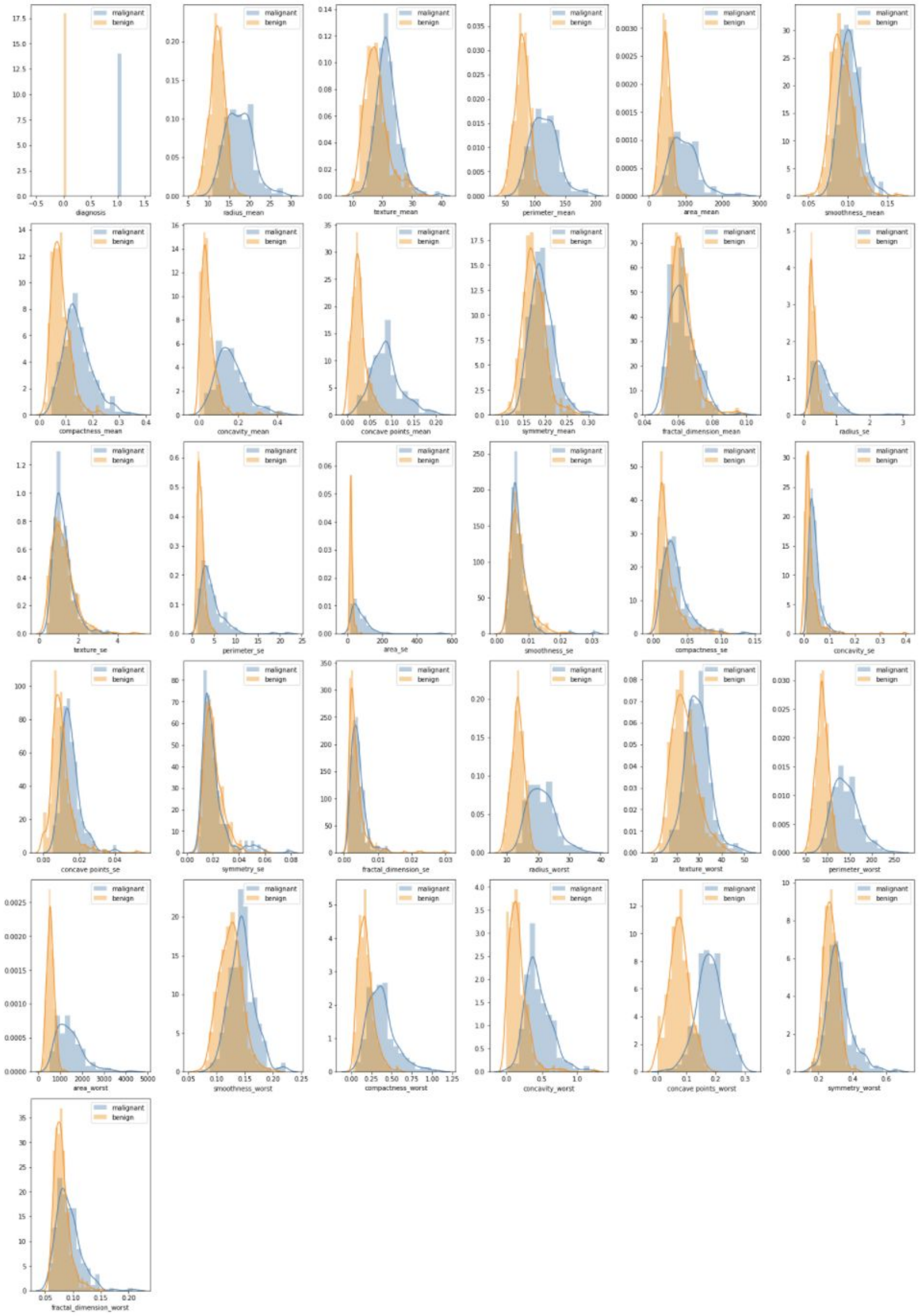
	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean
0	1.0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419
1	1.0	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812
2	1.0	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069
3	1.0	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597
4	1.0	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809

5 rows × 31 columns

```
# move diagnostic result into a new dataframe
diagnostic_result = pd.DataFrame()
diagnostic_result['diagnosis'] = data.iloc[:,0]
```

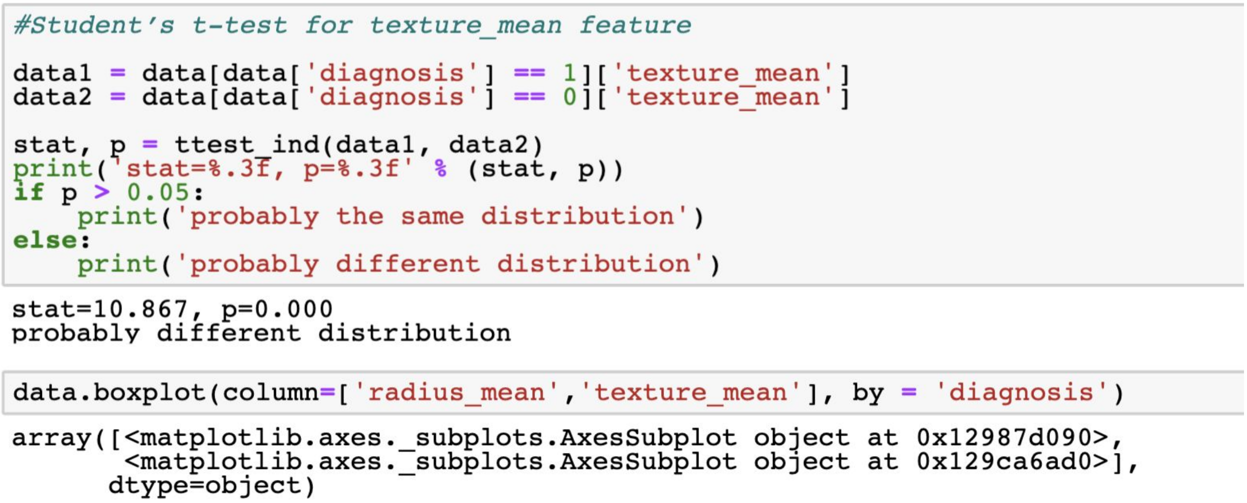
```
fig = plt.figure(figsize = (20,30))
j = 0
for i in data.columns:
    plt.subplot(6, 6, j+1)
    j += 1
    sns.distplot(data[i][diagnostic_result['diagnosis']== 1], color = 'steelblue', label = 'malignant')
    sns.distplot(data[i][diagnostic_result['diagnosis']== 0], color = 'darkorange', label ='benign')
    plt.legend(loc = 'best')
fig.suptitle('Breast cancer feature distribution')
fig.tight_layout()
fig.subplots_adjust (top=0.95)
plt.show()
```

1. Results:


malignant
benign
diagnosis
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst

2. There are many outliers in texture of mass compared to the mean radius of mass. The size of mass (radius_mean) will be a better criteria for malignancy diagnosis compared to the texture of mass.

In order to examine the spread of variables, boxplots were used to check skewness in the data. Box plots clearly show the outliers.
Student t-test shows that texture_mean values in B group vs. M group is statistically different. But box plots show that there are many outliers in texture_mean variables compared to radius mean.

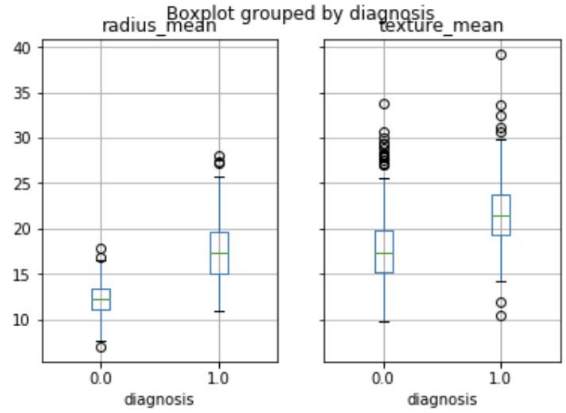


```python
#Student’s t-test for texture_mean feature

data1 = data[data['diagnosis'] == 1]['texture_mean']
data2 = data[data['diagnosis'] == 0]['texture_mean']

stat, p = ttest_ind(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('probably the same distribution')
else:
    print('probably different distribution')
```

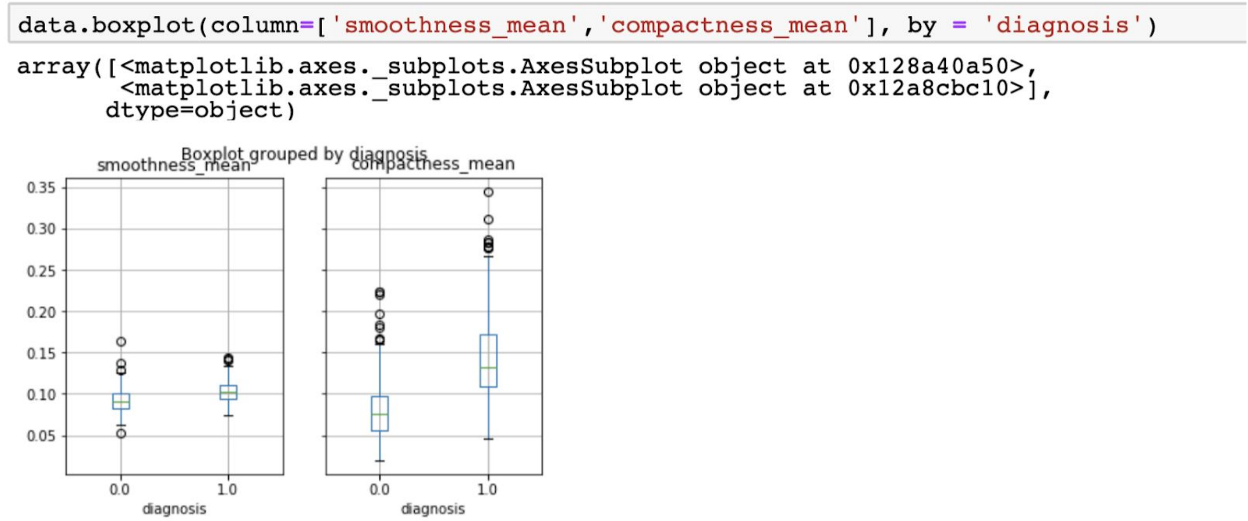
```
stat=10.867, p=0.000
probably different distribution
```

```python
data.boxplot(column=['radius_mean','texture_mean'], by = 'diagnosis')
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x12987d090>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x129ca6ad0>],
      dtype=object)
```



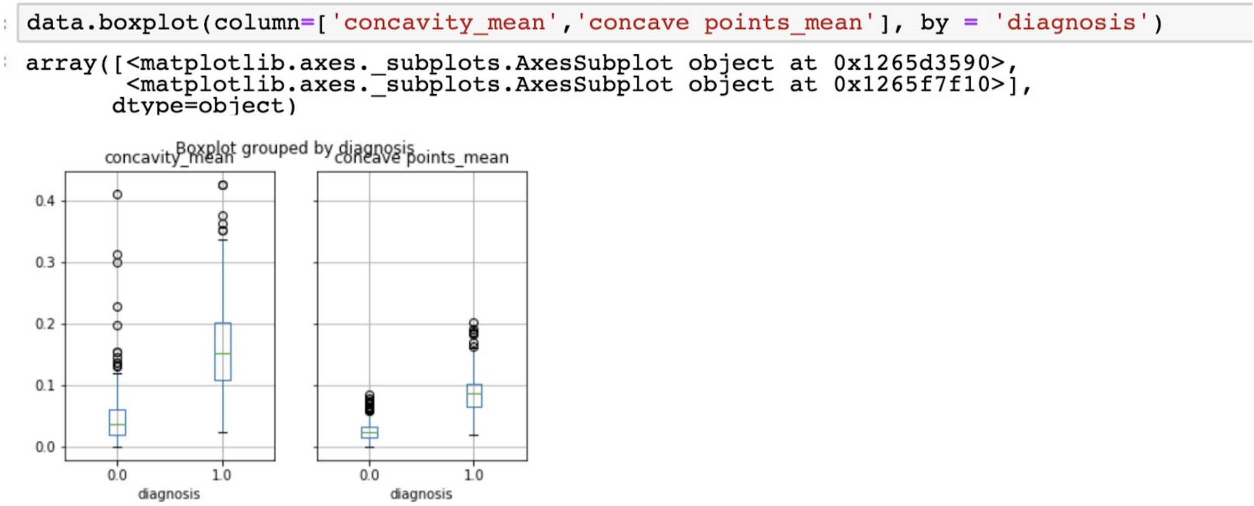
3.There is not much difference in smoothness of mass between groups. The compactness is higher in the malignancy group.



```
data.boxplot(column=['smoothness_mean','compactness_mean'], by = 'diagnosis')
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x128a40a50>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x12a8cbc10>],
      dtype=object)
```

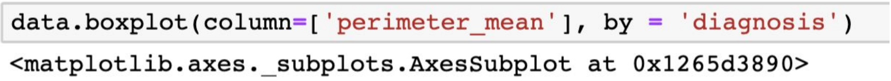
4. There are many outliers in concavity. Concave point is higher in the malignancy group.



```
data.boxplot(column=['concavity_mean','concave points_mean'], by = 'diagnosis')
```

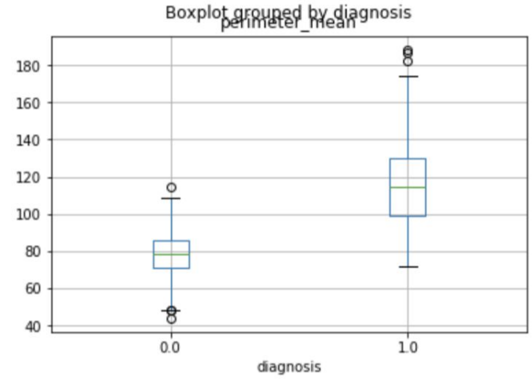
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x1265d3590>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x1265f7f10>],
      dtype=object)
```

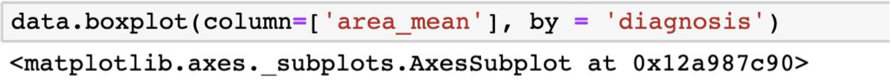
5. The perimeter and the area of mass are higher in the malignancy group.



```
data.boxplot(column=['perimeter_mean'], by = 'diagnosis')
```

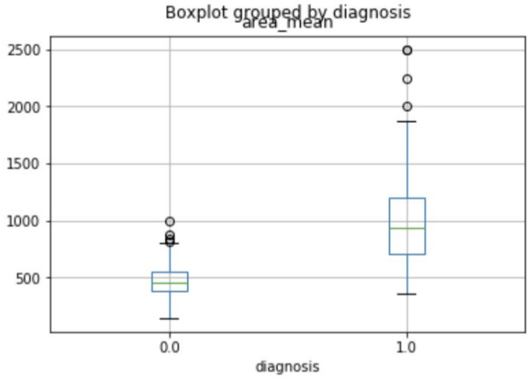
```
<matplotlib.axes._subplots.AxesSubplot at 0x1265d3890>
```



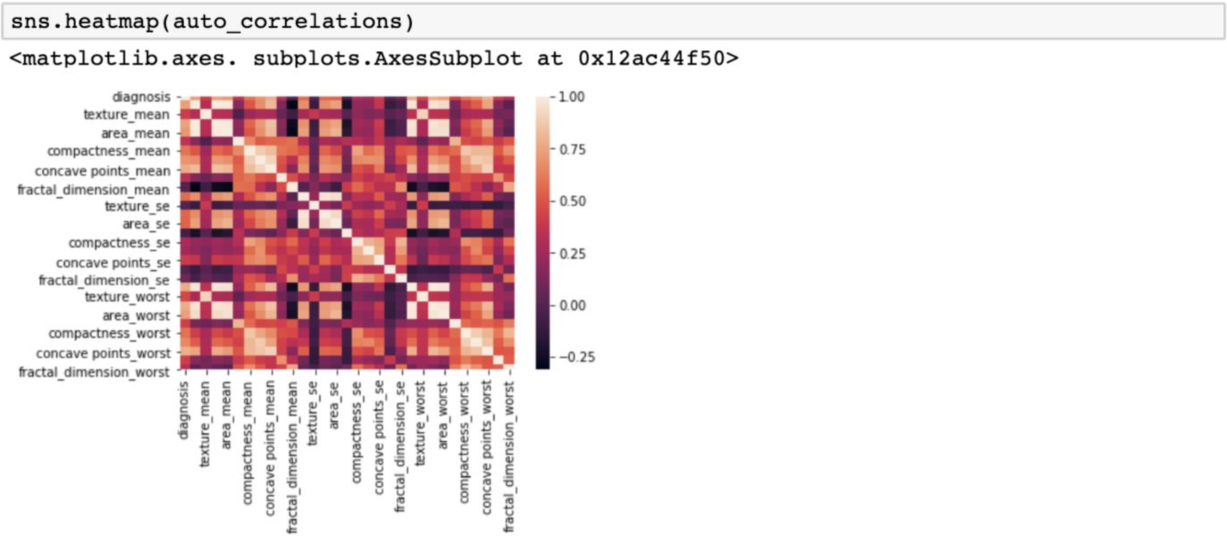


```
data.boxplot(column=['area_mean'], by = 'diagnosis')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x12a987c90>
```

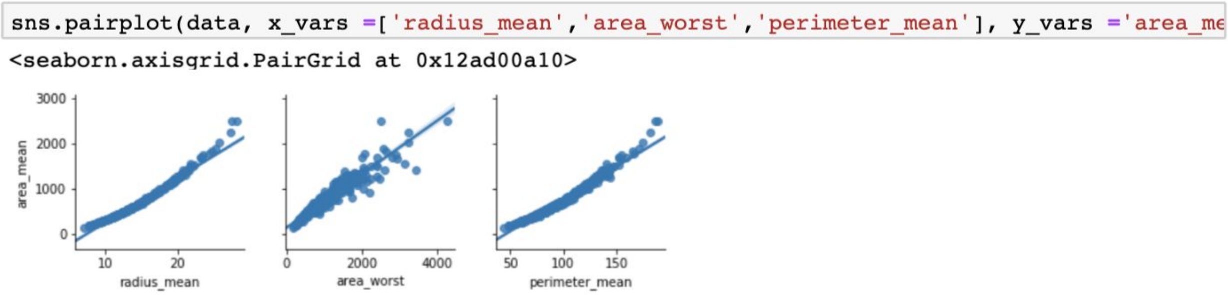


6. area_mean feature is highly correlated with radius_mean, area_worst and perimeter_mean.



```
sns.heatmap(auto_correlations)
```

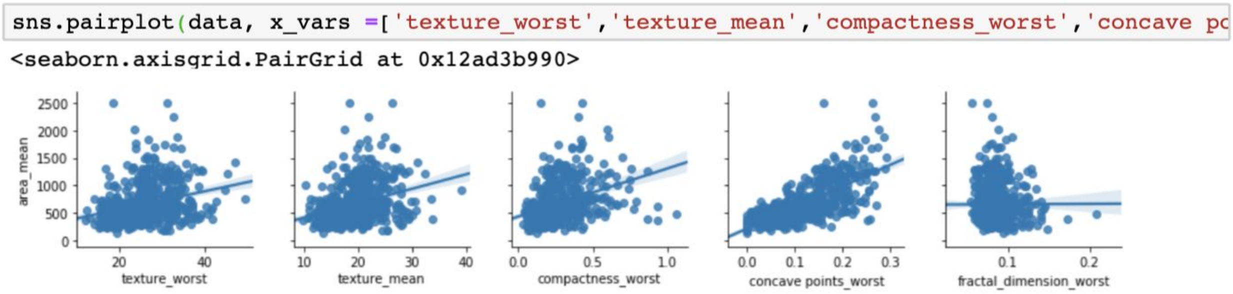
```
<matplotlib.axes. subplots.AxesSubplot at 0x12ac44f50>
```



```
sns.pairplot(data, x_vars =['radius_mean','area_worst','perimeter_mean'], y_vars ='area_me
```

```
<seaborn.axisgrid.PairGrid at 0x12ad00a10>
```

7. area_mean feature is not highly correlated with 'texture_worst','texture_mean','compactness_worst','concave points_worst' and 'fractal_dimension_worst'.



```
sns.pairplot(data, x_vars =['texture_worst','texture_mean','compactness_worst','concave po
```

```
<seaborn.axisgrid.PairGrid at 0x12ad3b990>
```