Exploratory Data Analysis-BreakHis Dataset

February 23,2019

[1] About Data

The dataset which we are going to use is BreakHis dataset caontainin 7909 histopathical breast cancer sample images from 82 patients respectively.
REPRESENTATION OF DATASET IN PROJECT IS AS FOLLOWS-

1. Cancer Class
1.1. Benign
This Class is represented by Integer-1
1.2. Malignant
This Class is represented by Integer-2

2. Cancer Type
2.1 Benign-A
Benign-A represents Adenosis.This Class is represented by Integer-11
2.2 Benign-FA
Benign-FA represents Fibro Adenoma.This Class is represented by Integer-12
2.3 Benign-TA
Benign-TA represents Tubulor Adenoma.This Class is represented by Integer-13
2.4 Benign-PT
Benign-PT represents Phyllodes Tumor.This Class is represented by Integer-14
2.5. Malignant-DC
Malignant-DC represents Ductol Carinoma.This Class is represented by Integer-21
2.6. Malignant-LC
Malignant-LC represents Lobular Carinoma.This Class is represented by Integer-22
2.7. Malignant-MC
Malignant-Mc represents Mucious Carinoma.This Class is represented by Integer-23
2.8. Malignant-PC
Malignant-PC represents Pappillary Carinoma.This Class is represented by Integer-24

3. Magnification
3.1. 40X - 40
3.2. 100X - 100
3.3. 200X - 200
3.4. 400X - 400

Note -

After Each visualization some counts are represented for elaborations of plots which are used for distribution.

Pre-Exploratory Data Analysis

Import Library

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sb
sb.set(style="darkgrid")
import matplotlib.pyplot as plt
```

Loading Numpy Array

In [2]:

```
# Train Arrays
data_cancerclass_train=np.load("train/data_cancerclass_train.npy")
data_cancertype_train=np.load("train/data_cancertype_train.npy")
data_mag_train=np.load("train/data_mag_train.npy")
# Test Arrays
data_cancerclass_test=np.load("test/data_cancerclass_test.npy")
data_cancertype_test=np.load("test/data_cancertype_test.npy")
data_mag_test=np.load("test/data_mag_test.npy")
```

[2] Train Arrays Visualization

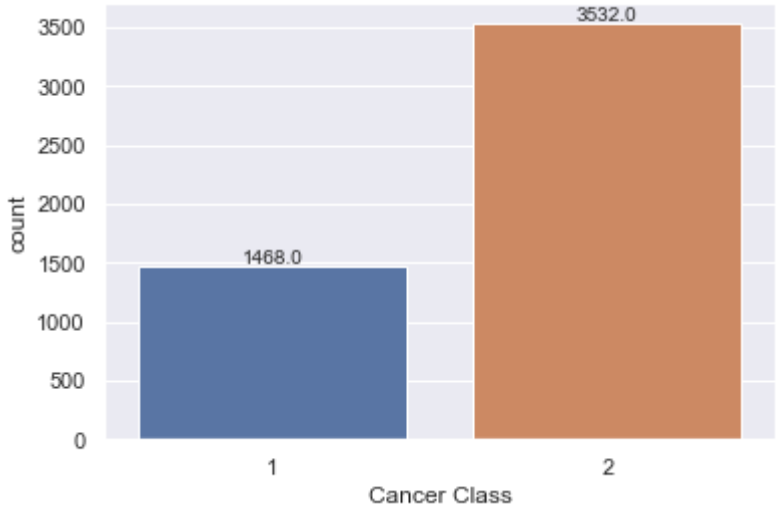
In [3]:

```
train_df=pd.DataFrame({'Cancer Class':data_cancerclass_train,
                       'Cancer Type':data_cancertype_train,
                       'Magnification':data_mag_train})
```

[2.1] Cancer Class

In [4]:

```
ax = sb.countplot(x="Cancer Class", data=train_df)
fig=ax.get_figure()
fig.savefig("Train Cancer Class.png")
for p in ax.patches:
    x=p.get_bbox().get_points()[:,0]
    y=p.get_bbox().get_points()[1,1]
    ax.annotate(y,(x.mean(), y),ha='center', va='bottom')
```



In [5]:

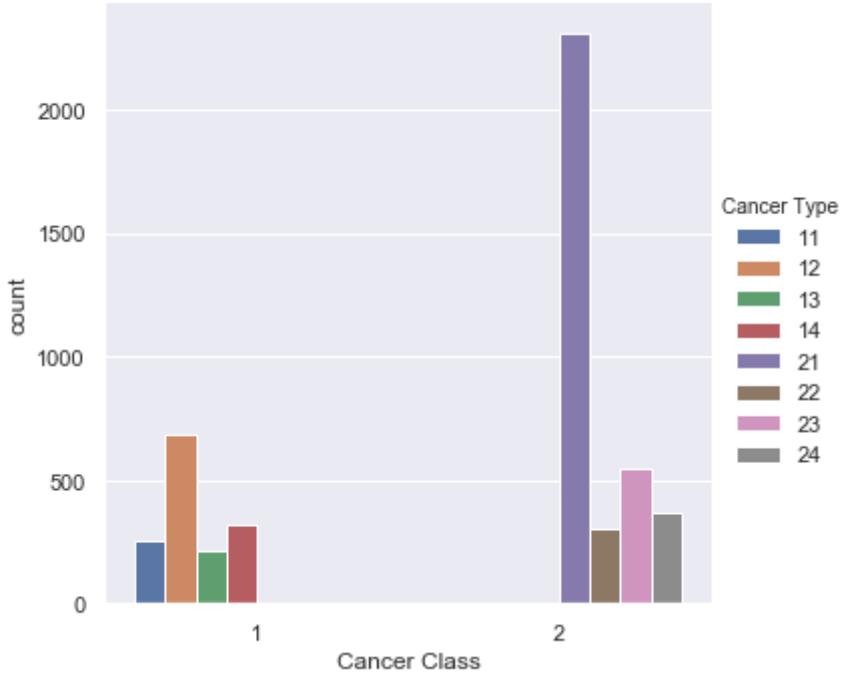
```
print(train_df.groupby("Cancer Class").count())
```

```
              Cancer Type  Magnification
Cancer Class
1                    1468           1468
2                    3532           3532
```

[2.3] Cancer Type

In [6]:

```
ax = sb.catplot(x="Cancer Class",hue="Cancer Type", data=train_df,kind="count")
ax.savefig("Train Cancer Class with cancer type.png")
```



In [7]:

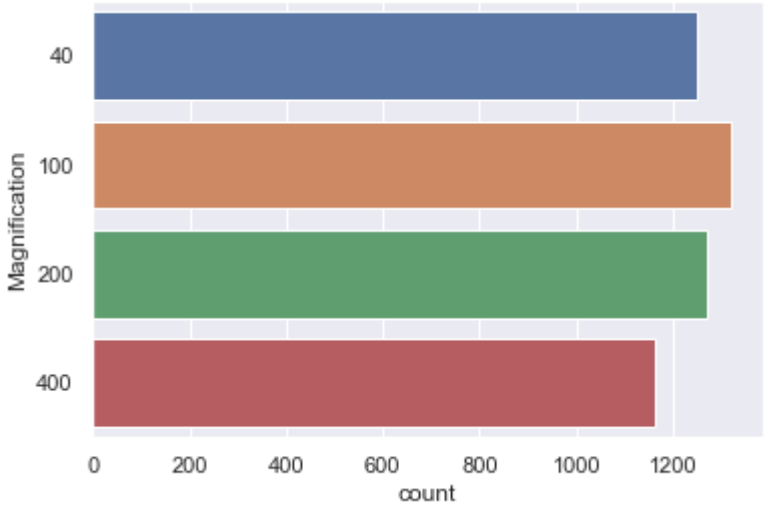
```
print(train_df.groupby("Cancer Type").count())
```

```
             Cancer Class  Magnification
Cancer Type
11                    252            252
12                    682            682
13                    217            217
14                    317            317
21                   2312           2312
22                    302            302
23                    547            547
24                    371            371
```

[2.3] Magnification

In [8]:

```
ax=sb.countplot(y="Magnification", data=train_df)
fig=ax.get_figure()
fig.savefig("Train Magnification.png")
```



In [9]:

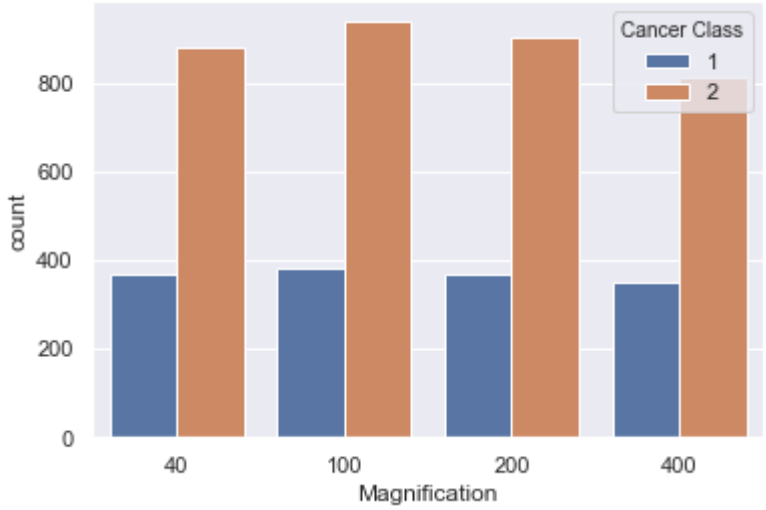
```
print(train_df.groupby("Magnification").count())
```

```
              Cancer Class  Cancer Type
Magnification
40                    1248         1248
100                   1320         1320
200                   1268         1268
400                   1164         1164
```

[2.4] Cancer Class Data Distribution

In [10]:

```
ax=sb.countplot(x="Magnification",hue="Cancer Class", data=train_df)
fig=ax.get_figure()
fig.savefig("Train Magnification in Train Numpy.png")
```



In [11]:

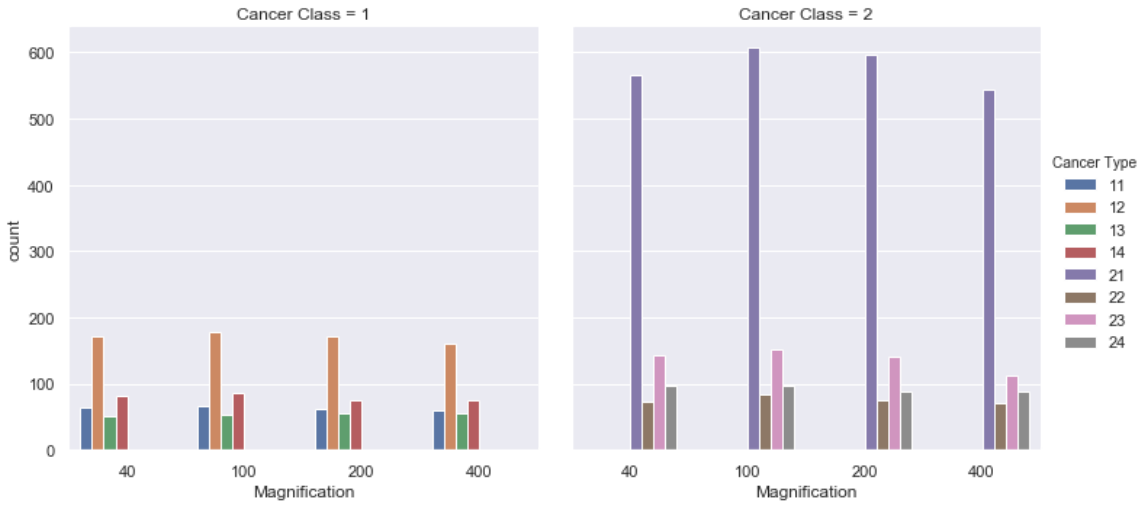
```
print(train_df.groupby(["Cancer Class","Magnification"]).count())
```

```
                              Cancer Type
Cancer Class Magnification
1            40                       369
             100                      382
             200                      367
             400                      350
2            40                       879
             100                      938
             200                      901
             400                      814
```

[2.5] Train Data Distribution

In [12]:

```
ax= sb.catplot(x="Magnification", hue="Cancer Type", col="Cancer Class",data=train_df,
kind="count")
ax.savefig("Train Cancer Type with Magnification using Cancer Class.png")
```



In [13]:

```
print(train_df.groupby(["Cancer Type","Magnification"]).count())
```

```
                          Cancer Class
Cancer Type Magnification
11          40                    64
            100                   66
            200                   63
            400                   59
12          40                   172
            100                  177
            200                  172
            400                  161
13          40                    51
            100                   54
            200                   56
            400                   56
14          40                    82
            100                   85
            200                   76
            400                   74
21          40                   565
            100                  607
            200                  596
            400                  544
22          40                    73
            100                   83
            200                   76
            400                   70
23          40                   143
            100                  151
            200                  141
            400                  112
24          40                    98
            100                   97
            200                   88
            400                   88
```

[3] Test Arrays Visualization

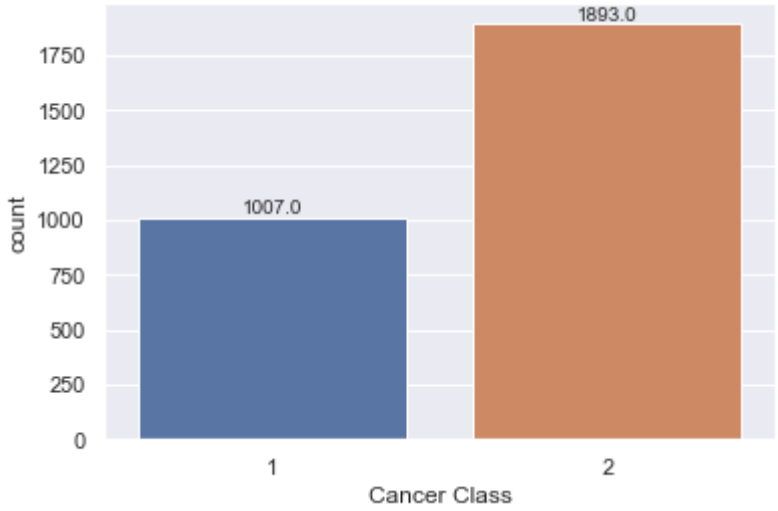
In [14]:

```
test_df=pd.DataFrame({'Cancer Class':data_cancerclass_test,
                      'Cancer Type':data_cancertype_test,
                      'Magnification':data_mag_test})
```

[3.1] Cancer Class

In [15]:

```
ax = sb.countplot(x="Cancer Class", data=test_df)
fig=ax.get_figure()
fig.savefig("Test Cancer Class.png")
for p in ax.patches:
    x=p.get_bbox().get_points()[:,0]
    y=p.get_bbox().get_points()[1,1]
    ax.annotate(y,(x.mean(), y),ha='center', va='bottom')
```



In [16]:

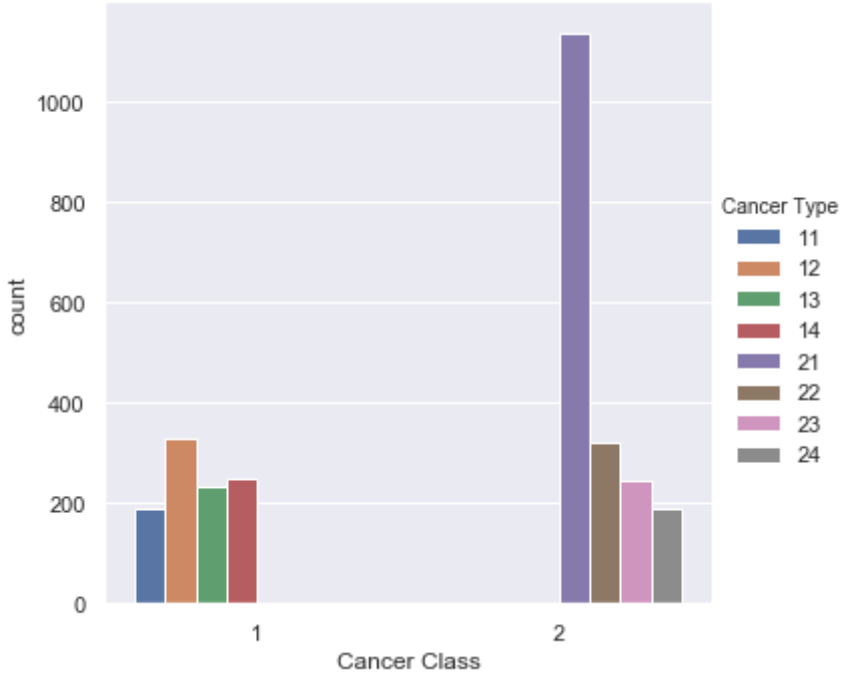
```
print(test_df.groupby("Cancer Class").count())
```

```
               Cancer Type  Magnification
Cancer Class
1                     1007           1007
2                     1893           1893
```

[3.2] Cancer Type

In [17]:

```
ax = sb.catplot(x="Cancer Class",hue="Cancer Type", data=test_df,kind="count")
ax.savefig("Test Cancer Class with cancer type.png")
```



In [18]:

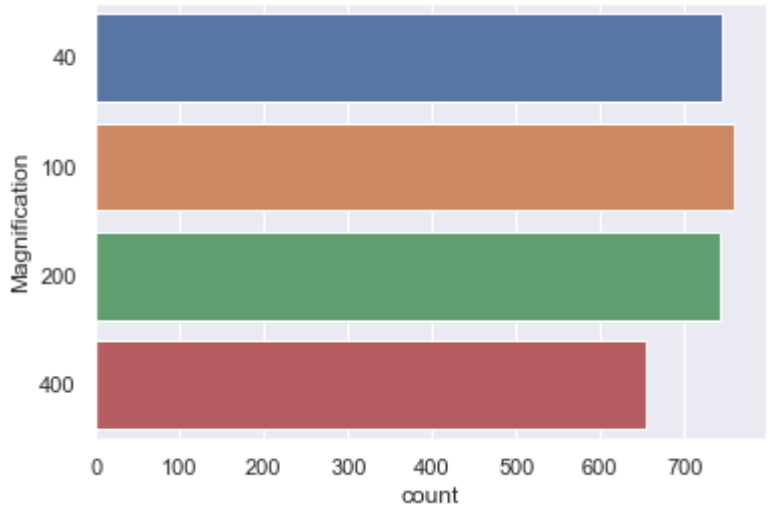
```
print(test_df.groupby("Cancer Type").count())
```

```
             Cancer Class  Magnification
Cancer Type
11                    191            191
12                    331            331
13                    235            235
14                    250            250
21                   1138           1138
22                    323            323
23                    244            244
24                    188            188
```

[3.3] Cancer Class Maginification

In [19]:

```
ax=sb.countplot(y="Magnification", data=test_df)
fig=ax.get_figure()
fig.savefig("Test Magnification.png")
```



In [20]:

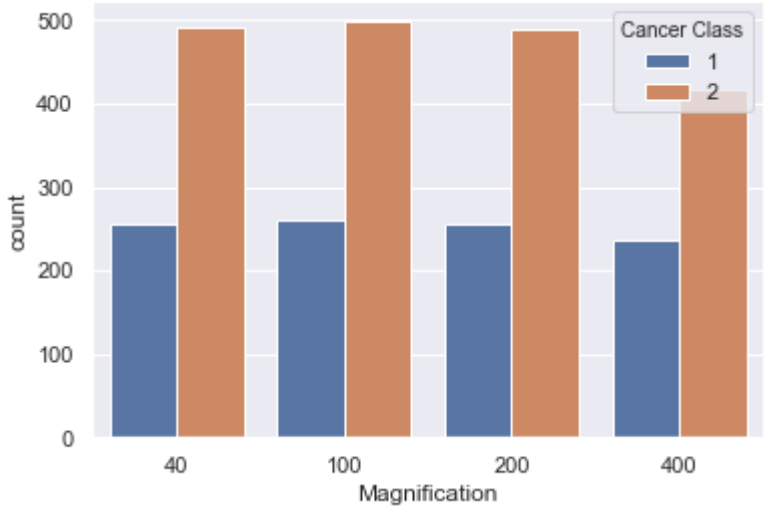
```
print(test_df.groupby("Magnification").count())
```

```
               Cancer Class  Cancer Type
Magnification
40                      745          745
100                     759          759
200                     743          743
400                     653          653
```

[3.4] Cancer Class Data Distribution

In [21]:

```
ax=sb.countplot(x="Magnification",hue="Cancer Class", data=test_df)
fig=ax.get_figure()
fig.savefig("Test Magnification in Test Numpy.png")
```



In [22]:

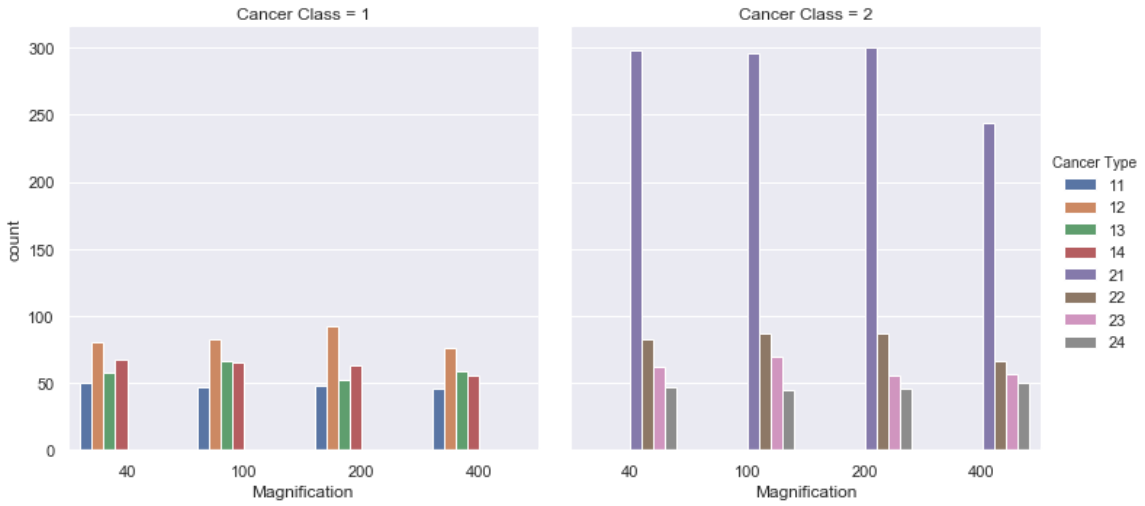
```
print(test_df.groupby(["Cancer Class","Magnification"]).count())
```

```
                              Cancer Type
Cancer Class Magnification
1            40                       255
             100                      261
             200                      255
             400                      236
2            40                       490
             100                      498
             200                      488
             400                      417
```

[3.5] Test Data Distribution

In [23]:

```
ax= sb.catplot(x="Magnification", hue="Cancer Type", col="Cancer Class",
...                 data=test_df, kind="count");
ax.savefig("Test Cancer Type with Magnification using Cancer Class.png")
```



In [24]:

```
print(test_df.groupby(["Cancer Type","Magnification"]).count())
```

```
                          Cancer Class
Cancer Type Magnification
11          40                  50
            100                 47
            200                 48
            400                 46
12          40                  80
            100                 83
            200                 92
            400                 76
13          40                  58
            100                 66
            200                 52
            400                 59
14          40                  67
            100                 65
            200                 63
            400                 55
21          40                 298
            100                296
            200                300
            400                244
22          40                  83
            100                 87
            200                 87
            400                 66
23          40                  62
            100                 70
            200                 55
            400                 57
24          40                  47
            100                 45
            200                 46
            400                 50
```

Post-Exploratory Data Analysis

After the dataset is retrived, it was passed through some Deep-Learning Algorithms for feature Extraction.The Algorithms are known as Deep Convolution Neural Networks.

The Used CNN's are as follows

1. VGG16
2. VGG19
3. Xception
4. ResNet50
5. InceptionV3
6. InceptionResNetV2

The Dataset was distributed as 5000 Train Samples,2900 Test Samples and Randomly 9 Images were removed for the checking of model.