1. **Research Question** Brain cancer is the most common cancer diagnosis in women and the second common cause of death among women in the world. What are the most significant and impactful information can be retrieved in the diagnosis features. How can predict Breast Cancer from the available data points. According to the UC Irving Machine Learning Repository, Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.
2. **Justification** This subject is relevant to me because I want to become a Data Scientist in a Healthcare Setting. I work a lot with medical claim data and research clinical-related datasets. This is the kind of project I would like work on because the impact my work could do in the oncology department. Machine Learning can help identify patients who are most likely to develop breast cancer based on their symptoms and lab results. Therefore, we can create specific care plans to improve those patients' overall health.
3. **Data Sources** The Breast Cancer Dataset is available in UC Irving Machine Learning Repository in the link below: https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic More information will be revealed during the Project regrading Features information and descriptions.
4. **Libraries** Pandas Library: Used for data analysis and loaded the dataset Numpy: Used for numerical data analysis and potentially setting seed for different machine learning algorithm. Sklearn Library: Efficient library used for machine leaning and statistical modeling including classification. Matplotlib and Seaborn: Python Libraries used for visualization for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.
5. **EDA and Summary Statistics** The Data Loaded directly via UC Irving Machine Learning Repository

```
# Install UC Irving Repo
pip install ucimlrepo
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.6-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.6
```

Start coding or generate with AI.

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
breast_cancer_wisconsin_diagnostic = fetch_ucirepo(id=17)

# data (as pandas dataframes)
X = breast_cancer_wisconsin_diagnostic.data.features
y = breast_cancer_wisconsin_diagnostic.data.targets

# metadata
print(breast_cancer_wisconsin_diagnostic.metadata)

# variable information
print(breast_cancer_wisconsin_diagnostic.variables)
```

```
{'uci_id': 17, 'name': 'Breast Cancer Wisconsin (Diagnostic)', 'repository_url'
                 name     role         type demographic description units  \
0                  ID       ID  Categorical        None        None  None
1           Diagnosis   Target  Categorical        None        None  None
2             radius1  Feature   Continuous        None        None  None
3            texture1  Feature   Continuous        None        None  None
4          perimeter1  Feature   Continuous        None        None  None
5               area1  Feature   Continuous        None        None  None
6         smoothness1  Feature   Continuous        None        None  None
7        compactness1  Feature   Continuous        None        None  None
```

TypeError TypeError TypeError Type •••

You

Please explain the error:

```
TypeError: 'NoneType' object is not subscriptable
```

Colab AI

The code is attempting to access the `columns` attribute of the `data` attribute of the `breast_cancer_wisconsin_diagnostic` object. However, the `data` attribute is `None`, so the code is trying to access the `columns` attribute of a `NoneType` object, which is not possible.

To fix the code, you need to check if the `data` attribute is not `None` before accessing its `columns` attribute. You can do this by using the following code:

```
if breast_cancer_wisconsin_diagnostic.data is not
    column_names = breast_cancer_wisconsin_diagnos
else:
    column_names = None  # or handle the case wher
```

Use code with caution

This code will check if the `data` attribute is not `None` before accessing its `columns` attribute. If the `data` attribute is `None`, it will set the `column_names` variable to `None` (or handle the case where the data is `None` in another way).

Rate this answer

```
8           concavity1  Feature   Continuous        None        None  None
9      concave_points1  Feature   Continuous        None        None  None
10           symmetry1  Feature   Continuous        None        None  None
11  fractal_dimension1  Feature   Continuous        None        None  None
12             radius2  Feature   Continuous        None        None  None
13            texture2  Feature   Continuous        None        None  None
14          perimeter2  Feature   Continuous        None        None  None
15               area2  Feature   Continuous        None        None  None
16         smoothness2  Feature   Continuous        None        None  None
17        compactness2  Feature   Continuous        None        None  None
18          concavity2  Feature   Continuous        None        None  None
19     concave_points2  Feature   Continuous        None        None  None
20           symmetry2  Feature   Continuous        None        None  None
21  fractal_dimension2  Feature   Continuous        None        None  None
22             radius3  Feature   Continuous        None        None  None
23            texture3  Feature   Continuous        None        None  None
24          perimeter3  Feature   Continuous        None        None  None
25               area3  Feature   Continuous        None        None  None
26         smoothness3  Feature   Continuous        None        None  None
27        compactness3  Feature   Continuous        None        None  None
28          concavity3  Feature   Continuous        None        None  None
29     concave_points3  Feature   Continuous        None        None  None
30           symmetry3  Feature   Continuous        None        None  None
31  fractal_dimension3  Feature   Continuous        None        None  None

   missing_values
0              no
1              no
2              no
3              no
4              no
5              no
6              no
7              no
8              no
9              no
10             no
11             no
12             no
13             no
14             no
15             no
16             no
17             no
18             no
19             no
20             no
```

During our analysis, we notice that the dataset has no missing values and there is only two categorical values. The dataset has 569 observations and 30 features.

```
# Breast cancer  Data
# Retreive the ids from the data sources for all the patients
id = breast_cancer_wisconsin_diagnostic.data.ids
# creatiing the full data set
breast_cancer = pd.concat([id, y,X], axis=1)


print(breast_cancer.head())
print("The shape of the dataset is :",breast_cancer.shape)
```

```
  Diagnosis  radius1  texture1  perimeter1   area1  smoothness1  compactness1  \
0         M    17.99     10.38      122.80  1001.0      0.11840       0.27760
1         M    20.57     17.77      132.90  1326.0      0.08474       0.07864
2         M    19.69     21.25      130.00  1203.0      0.10960       0.15990
3         M    11.42     20.38       77.58   386.1      0.14250       0.28390
4         M    20.29     14.34      135.10  1297.0      0.10030       0.13280

   concavity1  concave_points1  symmetry1  ...  radius3  texture3  perimeter3  \
0      0.3001          0.14710     0.2419  ...    25.38     17.33      184.60
1      0.0869          0.07017     0.1812  ...    24.99     23.41      158.80
2      0.1974          0.12790     0.2069  ...    23.57     25.53      152.50
3      0.2414          0.10520     0.2597  ...    14.91     26.50       98.87
4      0.1980          0.10430     0.1809  ...    22.54     16.67      152.20

    area3  smoothness3  compactness3  concavity3  concave_points3  symmetry3  \
0  2019.0       0.1622        0.6656      0.7119           0.2654     0.4601
1  1956.0       0.1238        0.1866      0.2416           0.1860     0.2750
2  1709.0       0.1444        0.4245      0.4504           0.2430     0.3613
3   567.7       0.2098        0.8663      0.6869           0.2575     0.6638
```

```
4  1575.0       0.1374        0.2050      0.4000           0.1625     0.2364

   fractal_dimension3
0             0.11890
1             0.08902
2             0.08758
3             0.17300
4             0.07678

[5 rows x 31 columns]
(569, 31)
```

```
breast_cancer.describe()
```

	radius1	texture1	perimeter1	area
count	569.000000	569.000000	569.000000	569.00000
mean	14.127292	19.289649	91.969033	654.88910
std	3.524049	4.301036	24.298981	351.91412
min	6.981000	9.710000	43.790000	143.50000
25%	11.700000	16.170000	75.170000	420.30000
50%	13.370000	18.840000	86.240000	551.10000
75%	15.780000	21.800000	104.100000	782.70000
max	28.110000	39.280000	188.500000	2501.00000

8 rows × 30 columns

There are about 357 malignant Diagnosis (M) and benign Diagnosis (B)

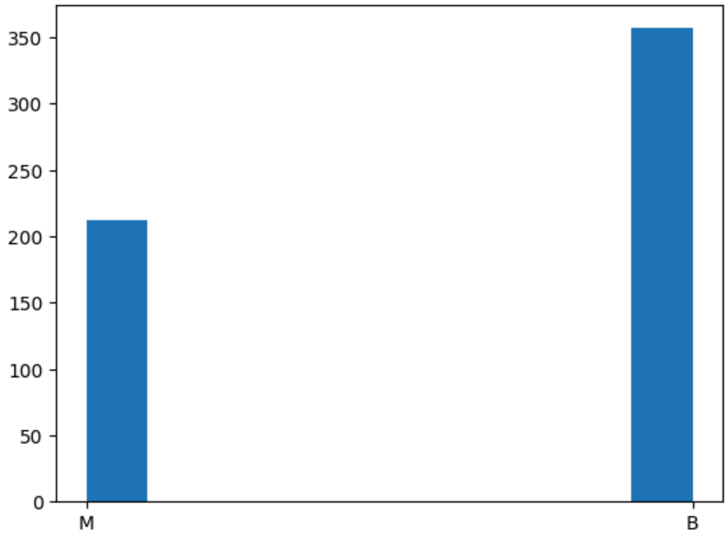
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
breast_cancer.Diagnosis.value_counts()
```

```
B    357
M    212
Name: Diagnosis, dtype: int64
```

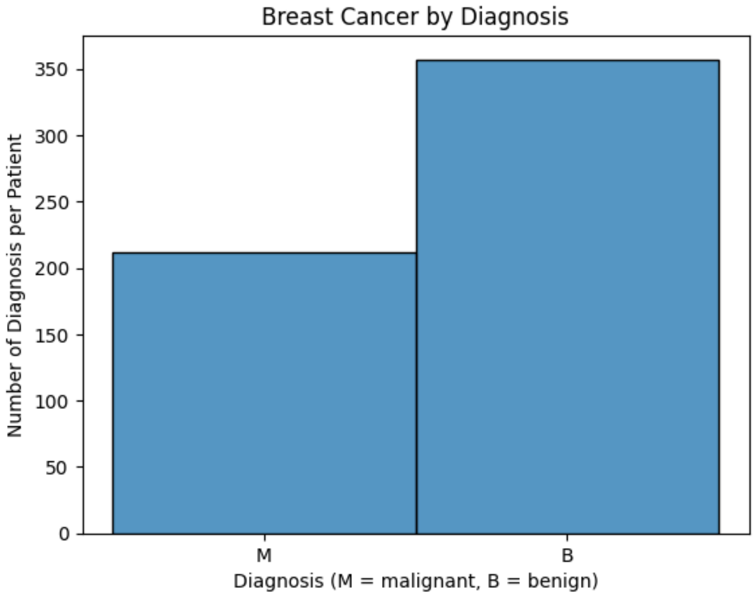
Double-click (or enter) to edit

```
# plt.figure(figsize=(7,12))
plt.hist(breast_cancer.Diagnosis)

plt.show()
```



```
sns.histplot(breast_cancer.Diagnosis, bins=10)  # Adjust the number of bins as needed
plt.title('Breast Cancer by Diagnosis')
plt.xlabel('Diagnosis (M = malignant, B = benign)')
plt.ylabel('Number of Diagnosis per Patient')
plt.show()
```



For our summary, It is ideal that use graph that help use see how diagnosis is distributed among our variables. Some of Features are well correlated with each other. During our analysis, a correlation graph will be implemented. Our next step will be to implemented the machine learning and evaulation techniques to get high accuracy model.

```
breast_cancer.corr() > 0.85
```

```
<ipython-input-124-65f0e27b2c3e>:1: FutureWarning: The default value of numeric_
  breast_cancer.corr() > 0.85
```

	radius1	texture1	perimeter1	area1	smoothness1	compactnes
radius1	True	False	True	True	False	Fa
texture1	False	True	False	False	False	Fa
perimeter1	True	False	True	True	False	Fa
area1	True	False	True	True	False	Fa
smoothness1	False	False	False	False	True	Fa
compactness1	False	False	False	False	False	Tr
concavity1	False	False	False	False	False	Tr
concave_points1	False	False	True	False	False	Fa
symmetry1	False	False	False	False	False	Fa
fractal_dimension1	False	False	False	False	False	Fa
radius2	False	False	False	False	False	Fa
texture2	False	False	False	False	False	Fa
perimeter2	False	False	False	False	False	Fa
area2	False	False	False	False	False	Fa
smoothness2	False	False	False	False	False	Fa
compactness2	False	False	False	False	False	Fa
concavity2	False	False	False	False	False	Fa
concave_points2	False	False	False	False	False	Fa
symmetry2	False	False	False	False	False	Fa
fractal_dimension2	False	False	False	False	False	Fa
radius3	True	False	True	True	False	Fa
texture3	False	True	False	False	False	Fa
perimeter3	True	False	True	True	False	Fa
area3	True	False	True	True	False	Fa
smoothness3	False	False	False	False	False	Fa
compactness3	False	False	False	False	False	Tr
concavity3	False	False	False	False	False	Fa
concave_points3	False	False	False	False	False	Fa
symmetry3	False	False	False	False	False	Fa

```
# Assuming breast_cancer is your DataFrame
correlation_matrix = breast_cancer.corr()

# Filter the correlation matrix to only show correlations greater than 0.85
high_correlations = correlation_matrix[correlation_matrix > 0.85]

# Print or display the filtered correlations
print(high_correlations)
```

```
                     radius1  texture1  perimeter1     area1  smoothness1  \
radius1             1.000000       NaN    0.997855  0.987357          NaN
texture1                 NaN  1.000000         NaN       NaN          NaN
perimeter1          0.997855       NaN    1.000000  0.986507          NaN
area1               0.987357       NaN    0.986507  1.000000          NaN
smoothness1              NaN       NaN         NaN       NaN          1.0
compactness1             NaN       NaN         NaN       NaN          NaN
concavity1               NaN       NaN         NaN       NaN          NaN
concave_points1          NaN       NaN    0.850977       NaN          NaN
symmetry1                NaN       NaN         NaN       NaN          NaN
fractal_dimension1       NaN       NaN         NaN       NaN          NaN
```

```
radius2                  NaN       NaN         NaN       NaN          NaN
texture2                 NaN       NaN         NaN       NaN          NaN
perimeter2               NaN       NaN         NaN       NaN          NaN
area2                    NaN       NaN         NaN       NaN          NaN
smoothness2              NaN       NaN         NaN       NaN          NaN
compactness2             NaN       NaN         NaN       NaN          NaN
concavity2               NaN       NaN         NaN       NaN          NaN
concave_points2          NaN       NaN         NaN       NaN          NaN
symmetry2                NaN       NaN         NaN       NaN          NaN
fractal_dimension2       NaN       NaN         NaN       NaN          NaN
radius3             0.969539       NaN    0.969476  0.962746          NaN
texture3                 NaN  0.912045         NaN       NaN          NaN
perimeter3          0.965137       NaN    0.970387  0.959120          NaN
area3               0.941082       NaN    0.941550  0.959213          NaN
smoothness3              NaN       NaN         NaN       NaN          NaN
compactness3             NaN       NaN         NaN       NaN          NaN
concavity3               NaN       NaN         NaN       NaN          NaN
concave_points3          NaN       NaN         NaN       NaN          NaN
symmetry3                NaN       NaN         NaN       NaN          NaN
fractal_dimension3       NaN       NaN         NaN       NaN          NaN

                    compactness1  concavity1  concave_points1  symmetry1  \
radius1                      NaN         NaN              NaN        NaN
texture1                     NaN         NaN              NaN        NaN
perimeter1                   NaN         NaN         0.850977        NaN
area1                        NaN         NaN              NaN        NaN
smoothness1                  NaN         NaN              NaN        NaN
compactness1            1.000000    0.883121              NaN        NaN
concavity1              0.883121    1.000000         0.921391        NaN
concave_points1              NaN    0.921391         1.000000        NaN
symmetry1                    NaN         NaN              NaN        1.0
fractal_dimension1           NaN         NaN              NaN        NaN
radius2                      NaN         NaN              NaN        NaN
texture2                     NaN         NaN              NaN        NaN
perimeter2                   NaN         NaN              NaN        NaN
```

Enter a prompt here

0 / 400

Responses may display inaccurate or offensive information that doesn't represent Google's views. Learn more