Thyroid Cancer Prediction

Objective:

Build a system that can predict if a Thyroid Cancer survivor can relapse(his or her cancer reoccurs

---

1. Problem Statement

The task is to do EDA on dataset and build a model to predict thyroid disease based on the features provided. The challenge is to create a model that can accurately predict the outcome.

---

2. Data Pre-Processing

2.1 Data Inspection and Summary Statistics

- **Load the Dataset:** Import the dataset and review its basic structure, including column names, data types, and a few initial records.
- **Generate Summary Statistics:** Calculate key statistics (mean, median, min, max, standard deviation, etc.) to understand the primary characteristics of each column.
- Changing column names and data types

2.2 Data Cleaning and Feature Engineering

- **Missing Values:** Check and handle missing values if present.
- **Duplicate Values:** Check duplicate values and handle if present.

2.3 Outlier Treatment

- **Outlier Detection:** Identify outliers in features box plots or Z-scores and apply treatment if necessary.

---

3. Exploratory Data Analysis (EDA)

3.1 Univariate Analysis

- **Numerical Data:** Visualize distributions with histograms and box plots.
- **Categorical Data:** Use bar charts to observe the distribution of the outcome variable.

3.2 Bivariate Analysis

- Create scatter plots to observe relationships between numerical features.

- Use box plots to explore how numerical features differ based on the outcome variable.

3.3 Multivariate Analysis

- Generate a heatmap of the correlation matrix to identify potential relationships.

---

4. Model Building

4.1 Encoding Categorical Variables:

- Convert the Categorical columns to binary format

4.2 Feature Engineering

- This step involves transforming raw data into meaningful features and outcome

4.3 Model Training

- Split the dataset into training and testing sets.
- Scalling the data
- Use a Logistic Regression to train the model on the training data.
- Model Evaluation
- Visualize the result

5. Advanced Modeling:

- Experiment with more complex models like RandomForest to improve predictions.

Import Libraries

```
import pandas as pd
import numpy as np
import  seaborn as sns
import plotly.express as px
import matplotlib.pyplot  as  plt
from sklearn.preprocessing import
OneHotEncoder,MinMaxScaler,LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression,SGDClassifier
from sklearn.ensemble import
RandomForestClassifier,AdaBoostClassifier,GradientBoostingClassifier,B
aggingClassifier
from sklearn.svm import SVC
from xgboost import XGBRFClassifier
from sklearn.metrics import
classification_report,confusion_matrix,ConfusionMatrixDisplay,accuracy
_score
import joblib
```

```
import warnings # ignore warnings

warnings.filterwarnings('ignore')
```

```
df = pd.read_csv('C://Users//PC\Downloads//Projects-20240722T093004Z-
001//Projects//thyroid_cancer//thyroid_cancer//dataset.csv')
```

```
df.head() # First 5 records
```

```
   Age Gender Smoking Hx Smoking Hx Radiothreapy Thyroid Function  \
0   27      F      No         No              No        Euthyroid
1   34      F      No        Yes              No        Euthyroid
2   30      F      No         No              No        Euthyroid
3   62      F      No         No              No        Euthyroid
4   62      F      No         No              No        Euthyroid

          Physical Examination Adenopathy       Pathology     Focality
Risk  \
0   Single nodular goiter-left         No  Micropapillary    Uni-Focal
Low
1          Multinodular goiter         No  Micropapillary    Uni-Focal
Low
2  Single nodular goiter-right         No  Micropapillary    Uni-Focal
Low
3  Single nodular goiter-right         No  Micropapillary    Uni-Focal
Low
4          Multinodular goiter         No  Micropapillary  Multi-Focal
Low

     T   N   M Stage       Response Recurred
0  T1a  N0  M0     I  Indeterminate       No
1  T1a  N0  M0     I      Excellent       No
2  T1a  N0  M0     I      Excellent       No
3  T1a  N0  M0     I      Excellent       No
4  T1a  N0  M0     I      Excellent       No
```

```
df.tail() # Last 5 records
```

```
     Age Gender Smoking Hx Smoking Hx Radiothreapy          Thyroid
Function  \
378   72      M     Yes         Yes             Yes
Euthyroid
379   81      M     Yes          No             Yes
Euthyroid
```

```
380   72      M     Yes        Yes               No
Euthyroid
381   61      M     Yes        Yes              Yes  Clinical
Hyperthyroidism
382   67      M     Yes         No               No
Euthyroid

            Physical Examination Adenopathy     Pathology     Focality
Risk  \
378  Single nodular goiter-right      Right     Papillary    Uni-Focal
High
379          Multinodular goiter  Extensive     Papillary  Multi-Focal
High
380          Multinodular goiter  Bilateral     Papillary  Multi-Focal
High
381          Multinodular goiter  Extensive  Hurthel cell  Multi-Focal
High
382          Multinodular goiter  Bilateral     Papillary  Multi-Focal
High

       T    N   M Stage                Response Recurred
378  T4b  N1b  M1   IVB  Biochemical Incomplete      Yes
379  T4b  N1b  M1   IVB   Structural Incomplete      Yes
380  T4b  N1b  M1   IVB   Structural Incomplete      Yes
381  T4b  N1b  M0   IVA   Structural Incomplete      Yes
382  T4b  N1b  M0   IVA   Structural Incomplete      Yes
```

2 Data Preprocessing¶

2.1 Data Inspection and Summary Statistics

```
df.shape # rews and col.
```

```
(383, 17)
```

```
df.ndim # dimentionality of data
```

```
2
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 383 entries, 0 to 382
Data columns (total 17 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Age                   383 non-null    int64
 1   Gender                383 non-null    object
 2   Smoking               383 non-null    object
 3   Hx Smoking            383 non-null    object
 4   Hx Radiothreapy       383 non-null    object
 5   Thyroid Function      383 non-null    object
 6   Physical Examination  383 non-null    object
 7   Adenopathy            383 non-null    object
 8   Pathology             383 non-null    object
 9   Focality              383 non-null    object
 10  Risk                  383 non-null    object
 11  T                     383 non-null    object
 12  N                     383 non-null    object
 13  M                     383 non-null    object
 14  Stage                 383 non-null    object
 15  Response              383 non-null    object
 16  Recurred              383 non-null    object
dtypes: int64(1), object(16)
memory usage: 51.0+ KB
```

```
df.describe() # Description of data
```

```
              Age
count  383.000000
mean    40.866841
std     15.134494
min     15.000000
25%     29.000000
50%     37.000000
75%     51.000000
max     82.000000
```

```
df.size # Total no. of elements
```

```
6511
```

2.2 Data Cleaning

Renaming columns¶

```
# rename
df.rename(columns={'T':'Tumor','N':'Nodal','M':'Metastasis'},inplace=True)
```

```
df.columns
```

```
Index(['Age', 'Gender', 'Smoking', 'Hx Smoking', 'Hx Radiothreapy',
       'Thyroid Function', 'Physical Examination', 'Adenopathy',
'Pathology',
       'Focality', 'Risk', 'Tumor', 'Nodal', 'Metastasis', 'Stage',
'Response',
       'Recurred'],
      dtype='object')
```

Missing Values

```
df.isnull().sum()
```

```
Age                     0
Gender                  0
Smoking                 0
Hx Smoking              0
Hx Radiothreapy         0
Thyroid Function        0
Physical Examination    0
Adenopathy              0
Pathology               0
Focality                0
Risk                    0
Tumor                   0
Nodal                   0
Metastasis              0
Stage                   0
Response                0
Recurred                0
dtype: int64
```

Duplicate Values

```
df.duplicated().sum()
```

```
19
```

```
df = df.drop_duplicates()
```

```
df.duplicated().sum()
```

```
0
```

There are no duplicate values

```
df.nunique()  # Unique values of columns
```

```
Age                     65
Gender                   2
Smoking                  2
Hx Smoking               2
Hx Radiothreapy          2
Thyroid Function         5
Physical Examination     5
Adenopathy               6
Pathology                4
Focality                 2
Risk                     3
Tumor                    7
Nodal                    3
Metastasis               2
Stage                    5
Response                 4
Recurred                 2
dtype: int64
```

2.3 Outlier Treatment

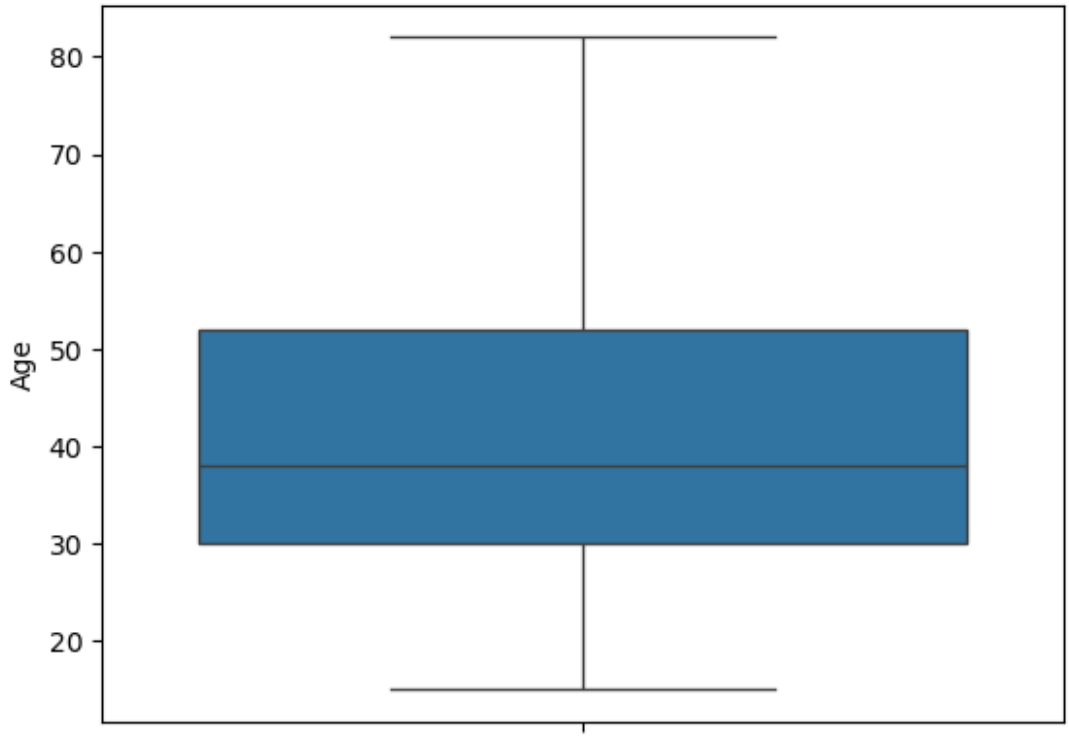
```
df.dtypes
```

```
Age                      int64
Gender                  object
Smoking                 object
Hx Smoking              object
Hx Radiothreapy         object
Thyroid Function        object
```

```
Physical Examination    object
Adenopathy              object
Pathology               object
Focality                object
Risk                    object
Tumor                   object
Nodal                   object
Metastasis              object
Stage                   object
Response                object
Recurred                object
dtype: object
```

```
sns.boxplot (data = df['Age'])
```

```
<Axes: ylabel='Age'>
```




```
# No Outlier
```

3. EDA

3.1 Univariate Analysis

Visualize individual variables to understand their distribution (e.g., histograms for numerical data, bar charts for categorical data).

3.2 Bivariate and Multivariate Analysis

Explore relationships between variables by visualizing pairs of variables or groups of variables (e.g., scatter plots, heatmaps).

```
```

3.1 Univariate Analysis

Age Analysis

```
fig = px.histogram(data_frame=df,
                   x ='Age',
                   nbins=20,marginal='box',

color_discrete_sequence=px.colors.sequential.GnBu_r,
                   text_auto=True,title='The Distribution of Age')
fig.update_layout()
fig.show()
```


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"alignmentgroup":"True","bingroup":"x","hovertemplate":"Age=%
{x}<br>count=%{y}<extra></extra>","legendgroup":"","marker":
{"color":"rgb(8,64,129)","pattern":
{"shape":""}},"name":"","nbinsx":20,"offsetgroup":"","orientation":"v"
,"showlegend":false,"texttemplate":"%{value}","type":"histogram","x":
[27,34,30,62,62,52,41,46,51,40,75,59,49,50,76,42,40,44,43,52,41,44,36,
70,60,33,43,26,41,37,37,30,55,52,37,31,43,34,45,20,38,38,33,31,31,26,2
9,43,30,25,27,25,21,43,23,23,43,24,35,54,54,22,38,51,22,40,69,31,29,30
,28,22,35,50,27,17,27,40,33,25,73,36,35,31,18,62,62,39,37,26,31,24,57,
28,44,42,27,51,33,42,26,24,60,60,31,66,44,32,26,37,33,23,47,28,28,44,3
1,27,56,63,24,30,31,28,51,20,21,42,28,41,42,49,29,29,25,41,33,27,50,19
,35,63,24,36,31,24,33,24,28,22,27,28,29,40,55,40,38,21,31,30,50,34,45,
52,67,72,45,45,67,56,30,50,35,23,44,23,26,61,31,68,57,27,25,20,33,36,3
6,40,17,24,38,28,36,50,51,55,31,33,28,48,40,29,20,35,56,20,62,17,21,20
,40,38,21,31,34,60,60,62,36,29,33,75,62,56,52,35,34,32,27,52,46,30,32,
25,38,31,37,21,34,30,48,31,52,38,41,41,70,19,41,32,35,39,45,46,45,28,3
1,81,41,56,47,37,32,53,30,34,62,58,55,21,27,46,44,29,26,42,56,51,61,42
,34,67,63,67,73,26,30,36,31,40,49,38,27,27,33,32,29,37,48,30,33,80,62,
63,60,79,65,35,58,34,56,52,51,31,44,15,29,53,45,38,48,42,23,22,44,31,2
```

```
5,32,82,58,68,37,59,21,73,35,32,54,26,53,35,49,34,80,67,68,71,64,80,56
,71,78,51,67,31,62,59,40,46,72,81,72,61,67],"xaxis":"x","yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"Age=%{x}<extra></
extra>","legendgroup":"","marker":
{"color":"rgb(8,64,129)"},"name":"","notched":true,"offsetgroup":"","s
howlegend":false,"type":"box","x":
[27,34,30,62,62,52,41,46,51,40,75,59,49,50,76,42,40,44,43,52,41,44,36,
70,60,33,43,26,41,37,37,30,55,52,37,31,43,34,45,20,38,38,33,31,31,26,2
9,43,30,25,27,25,21,43,23,23,43,24,35,54,54,22,38,51,22,40,69,31,29,30
,28,22,35,50,27,17,27,40,33,25,73,36,35,31,18,62,62,39,37,26,31,24,57,
28,44,42,27,51,33,42,26,24,60,60,31,66,44,32,26,37,33,23,47,28,28,44,3
1,27,56,63,24,30,31,28,51,20,21,42,28,41,42,49,29,29,25,41,33,27,50,19
,35,63,24,36,31,24,33,24,28,22,27,28,29,40,55,40,38,21,31,30,50,34,45,
52,67,72,45,45,67,56,30,50,35,23,44,23,26,61,31,68,57,27,25,20,33,36,3
6,40,17,24,38,28,36,50,51,55,31,33,28,48,40,29,20,35,56,20,62,17,21,20
,40,38,21,31,34,60,60,62,36,29,33,75,62,56,52,35,34,32,27,52,46,30,32,
25,38,31,37,21,34,30,48,31,52,38,41,41,70,19,41,32,35,39,45,46,45,28,3
1,81,41,56,47,37,32,53,30,34,62,58,55,21,27,46,44,29,26,42,56,51,61,42
,34,67,63,67,73,26,30,36,31,40,49,38,27,27,33,32,29,37,48,30,33,80,62,
63,60,79,65,35,58,34,56,52,51,31,44,15,29,53,45,38,48,42,23,22,44,31,2
5,32,82,58,68,37,59,21,73,35,32,54,26,53,35,49,34,80,67,68,71,64,80,56
,71,78,51,67,31,62,59,40,46,72,81,72,61,67],"xaxis":"x2","yaxis":"y2"}
],"layout":{"autosize":true,"barmode":"relative","legend":
{"tracegroupgap":0},"template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
```

```
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
```

```
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
```

{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"The Distribution of Age"},"xaxis":
{"anchor":"y","autorange":true,"domain":[0,1],"range":
[11.277777777777779,85.72222222222223],"title":
{"text":"Age"},"type":"linear"},"xaxis2":
{"anchor":"y2","autorange":true,"domain":[0,1],"matches":"x","range":
[11.277777777777779,85.72222222222223],"showgrid":true,"showticklabels
":false,"type":"linear"},"yaxis":
{"anchor":"x","autorange":true,"domain":[0,0.8316],"range":
[0,66.3157894736842],"title":{"text":"count"}},"yaxis2":
{"anchor":"x2","autorange":true,"domain":
[0.8416,1],"matches":"y2","range":[-
0.5,0.5],"showgrid":false,"showline":false,"showticklabels":false,"tic
ks":"","type":"category"}}}

```
fig = px.histogram(data_frame=df,
                   x ='Age',
                   nbins=20,marginal='box',
                   color='Recurred',

color_discrete_sequence=px.colors.sequential.Agsunset_r,
                   text_auto=True,title='The Distribution of Recurred
based on Age')
fig.update_layout()
fig.show()
```

{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"alignmentgroup":"True","bingroup":"x","hovertemplate":"Recurred=No<
br>Age=%{x}
count=%{y}<extra></extra>","legendgroup":"No","marker":
{"color":"rgb(237, 217, 163)","pattern":
{"shape":""}},"name":"No","nbinsx":20,"offsetgroup":"No","orientation"
:"v","showlegend":true,"texttemplate":"%
{value}","type":"histogram","x":
[27,34,30,62,62,52,41,46,51,40,75,59,49,50,76,42,40,44,43,52,41,44,36,
70,60,33,43,26,41,37,37,30,55,52,37,31,43,34,45,20,38,38,33,31,31,29,4
3,30,25,27,25,21,43,23,23,43,24,35,54,54,22,38,51,22,40,69,31,29,30,28
,22,35,50,27,17,27,40,33,25,73,62,39,37,26,31,24,57,28,44,42,27,51,33,
42,26,24,60,60,31,66,44,32,26,37,33,23,47,28,28,44,31,27,56,63,24,30,3
1,28,51,20,21,42,28,41,42,49,29,29,25,41,33,27,50,19,35,63,24,36,31,24
,33,24,28,22,27,28,29,40,55,40,38,21,31,30,50,34,45,52,67,72,45,45,67,
56,30,50,35,23,44,23,26,61,31,68,57,27,25,20,33,36,36,40,17,24,38,28,3
6,50,51,55,31,33,28,48,40,29,20,35,52,35,34,32,27,52,46,30,32,25,38,31
,37,21,34,30,48,31,52,38,41,41,70,19,41,32,35,39,45,46,45,28,31,81,41,
56,47,37,32,53,30,34,62,58,55,21,27,46,44,29,26,42,56,51,61,48,42,32],
"xaxis":"x","yaxis":"y"},
{"alignmentgroup":"True","hovertemplate":"Recurred=No
Age=%
{x}<extra></extra>","legendgroup":"No","marker":{"color":"rgb(237,
217,

```
163)"},"name":"No","notched":true,"offsetgroup":"No","showlegend":fals
e,"type":"box","x":
[27,34,30,62,62,52,41,46,51,40,75,59,49,50,76,42,40,44,43,52,41,44,36,
70,60,33,43,26,41,37,37,30,55,52,37,31,43,34,45,20,38,38,33,31,31,29,4
3,30,25,27,25,21,43,23,23,43,24,35,54,54,22,38,51,22,40,69,31,29,30,28
,22,35,50,27,17,27,40,33,25,73,62,39,37,26,31,24,57,28,44,42,27,51,33,
42,26,24,60,60,31,66,44,32,26,37,33,23,47,28,28,44,31,27,56,63,24,30,3
1,28,51,20,21,42,28,41,42,49,29,29,25,41,33,27,50,19,35,63,24,36,31,24
,33,24,28,22,27,28,29,40,55,40,38,21,31,30,50,34,45,52,67,72,45,45,67,
56,30,50,35,23,44,23,26,61,31,68,57,27,25,20,33,36,36,40,17,24,38,28,3
6,50,51,55,31,33,28,48,40,29,20,35,52,35,34,32,27,52,46,30,32,25,38,31
,37,21,34,30,48,31,52,38,41,41,70,19,41,32,35,39,45,46,45,28,31,81,41,
56,47,37,32,53,30,34,62,58,55,21,27,46,44,29,26,42,56,51,61,48,42,32],
"xaxis":"x2","yaxis":"y2"},
{"alignmentgroup":"True","bingroup":"x","hovertemplate":"Recurred=Yes<
br>Age=%{x}<br>count=%{y}<extra></
extra>","legendgroup":"Yes","marker":{"color":"rgb(246, 169,
122)","pattern":
{"shape":""}},"name":"Yes","nbinsx":20,"offsetgroup":"Yes","orientatio
n":"v","showlegend":true,"texttemplate":"%
{value}","type":"histogram","x":
[26,36,35,31,18,62,56,20,62,17,21,20,40,38,21,31,34,60,60,62,36,29,33,
75,62,56,42,34,67,63,67,73,26,30,36,31,40,49,38,27,27,33,32,29,37,48,3
0,33,80,62,63,60,79,65,35,58,34,56,52,51,31,44,15,29,53,45,38,23,22,44
,31,25,32,82,58,68,37,59,21,73,35,54,26,53,35,49,34,80,67,68,71,64,80,
56,71,78,51,67,31,62,59,40,46,72,81,72,61,67],"xaxis":"x","yaxis":"y"}
,{"alignmentgroup":"True","hovertemplate":"Recurred=Yes<br>Age=%
{x}<extra></extra>","legendgroup":"Yes","marker":{"color":"rgb(246,
169,
122)"},"name":"Yes","notched":true,"offsetgroup":"Yes","showlegend":fa
lse,"type":"box","x":
[26,36,35,31,18,62,56,20,62,17,21,20,40,38,21,31,34,60,60,62,36,29,33,
75,62,56,42,34,67,63,67,73,26,30,36,31,40,49,38,27,27,33,32,29,37,48,3
0,33,80,62,63,60,79,65,35,58,34,56,52,51,31,44,15,29,53,45,38,23,22,44
,31,25,32,82,58,68,37,59,21,73,35,54,26,53,35,49,34,80,67,68,71,64,80,
56,71,78,51,67,31,62,59,40,46,72,81,72,61,67],"xaxis":"x2","yaxis":"y2
"}],"layout":{"autosize":true,"barmode":"relative","legend":{"title":
{"text":"Recurred"},"tracegroupgap":0},"template":{"data":{"bar":
[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
```

```
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
```

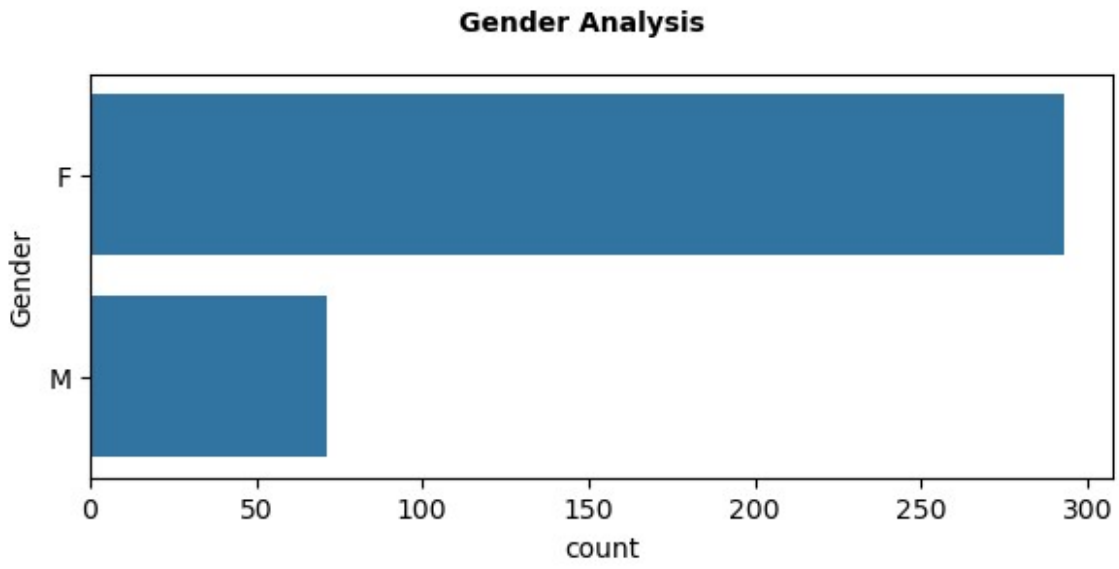
```
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
```

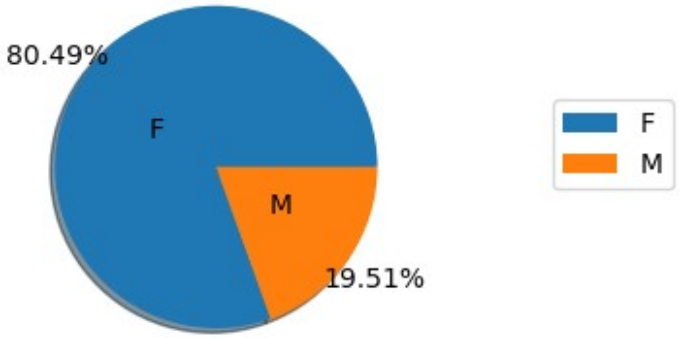
```
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"The Distribution of Recurred based on Age"},"xaxis":
{"anchor":"y","autorange":true,"domain":[0,1],"range":
[11.277777777777779,85.72222222222223],"title":
{"text":"Age"},"type":"linear"},"xaxis2":
{"anchor":"y2","autorange":true,"domain":[0,1],"matches":"x","range":
[11.277777777777779,85.72222222222223],"showgrid":true,"showticklabels
":false,"type":"linear"},"yaxis":
{"anchor":"x","autorange":true,"domain":[0,0.7326],"range":
[0,66.3157894736842],"title":{"text":"count"}},"yaxis2":
{"anchor":"x2","autorange":true,"domain":
[0.7426,1],"matches":"y2","range":[-
0.5,1.5],"showgrid":false,"showline":false,"showticklabels":false,"tic
ks":"","type":"category"}}}
```

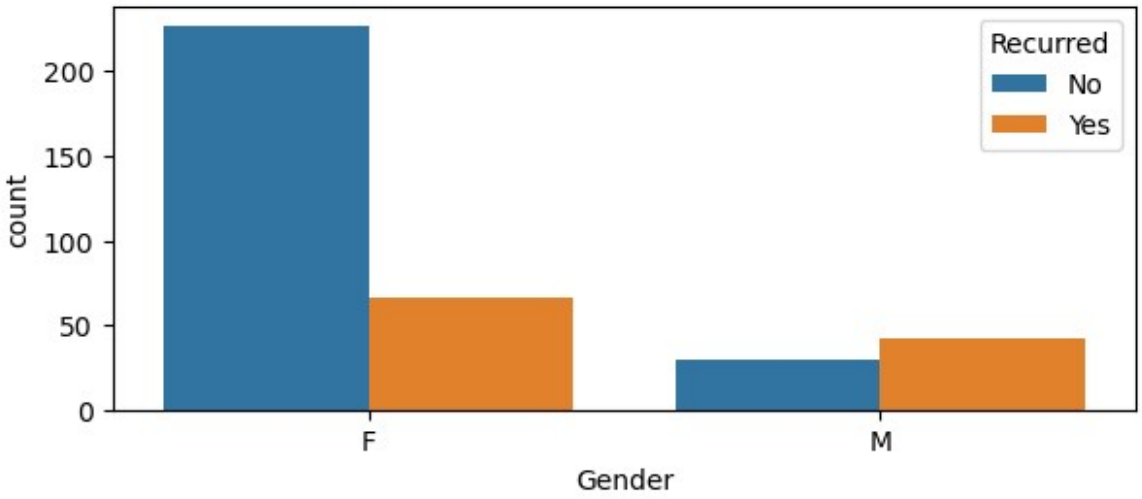

Gender Analysis

```
fig, ax = plt.subplots(3, 1, figsize=(6, 8))
fig.suptitle('Gender Analysis', fontsize=10, fontweight='bold')
plt.tight_layout()

count = df['Gender'].value_counts()

labels = df['Gender'].value_counts().index.tolist()

#Top ax
sns.countplot(y="Gender",data=df, ax=ax[0])
#middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
```

```
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Gender', hue='Recurred', data=df, ax=ax[2])
plt.show()
```

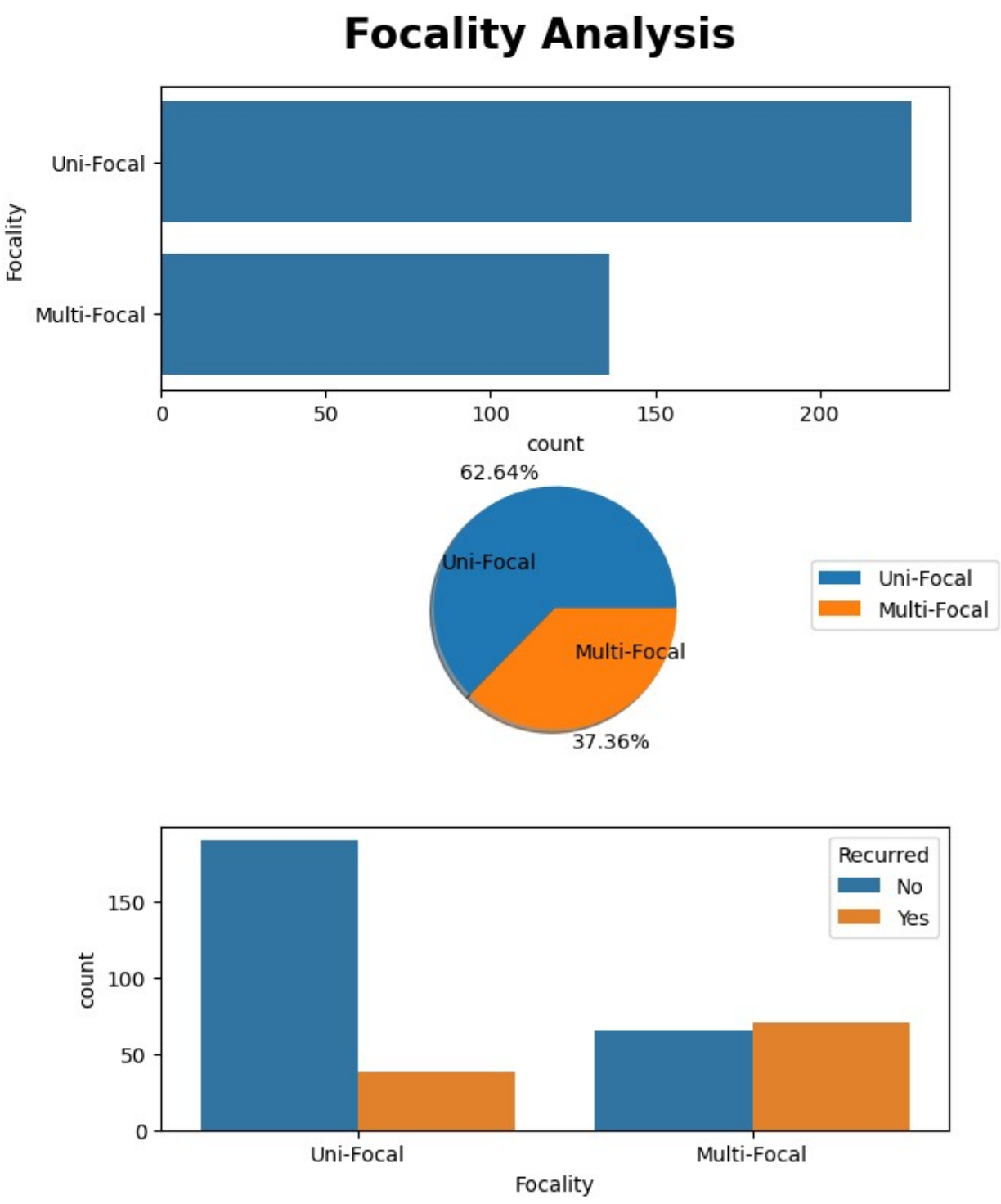

Gender Analysis
F
M
Gender
0
50
100
150
200
250
300
count


80.49%
F
M
19.51%
F
M


Recurred
No
Yes
200
150
100
50
0
count
F
M
Gender

Focality Analysis

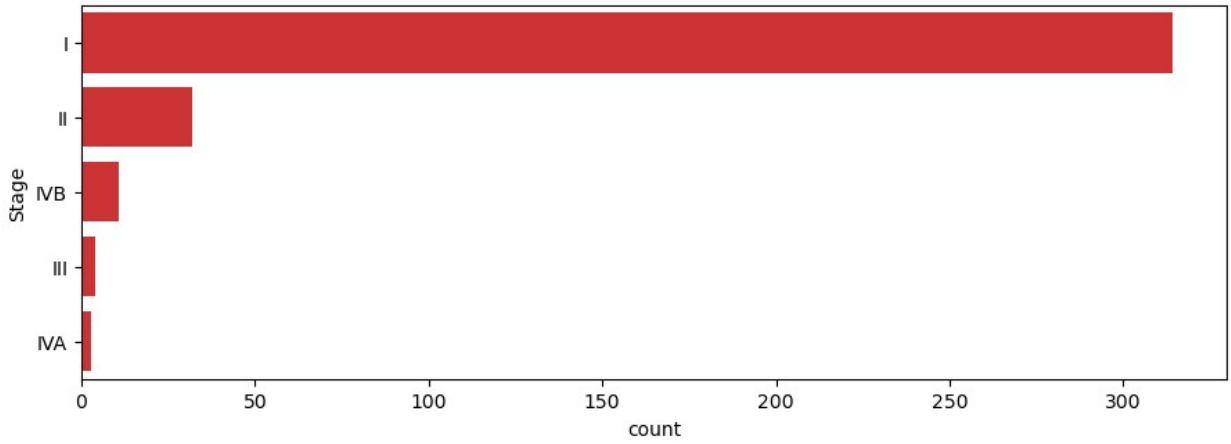
```
fig, ax = plt.subplots(3, 1, figsize=(6, 8))
fig.suptitle('Focality Analysis', fontsize=20, fontweight='bold')
plt.tight_layout()

count = df['Focality'].value_counts()

labels = df['Focality'].value_counts().index.tolist()

#Top ax
sns.countplot(y="Focality",data=df, ax=ax[0])
#middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Focality', hue='Recurred', data=df, ax=ax[2])
plt.show()
```

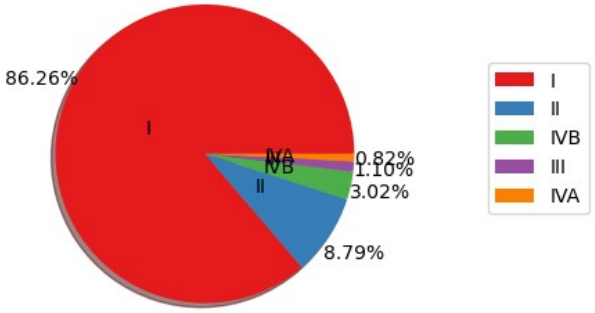

Focality Analysis
Uni-Focal
Multi-Focal
Focality
0
50
100
150
200
count
62.64%
Uni-Focal
Multi-Focal
37.36%
Uni-Focal
Multi-Focal
Recurred
No
Yes
150
100
50
0
count
Uni-Focal
Multi-Focal
Focality

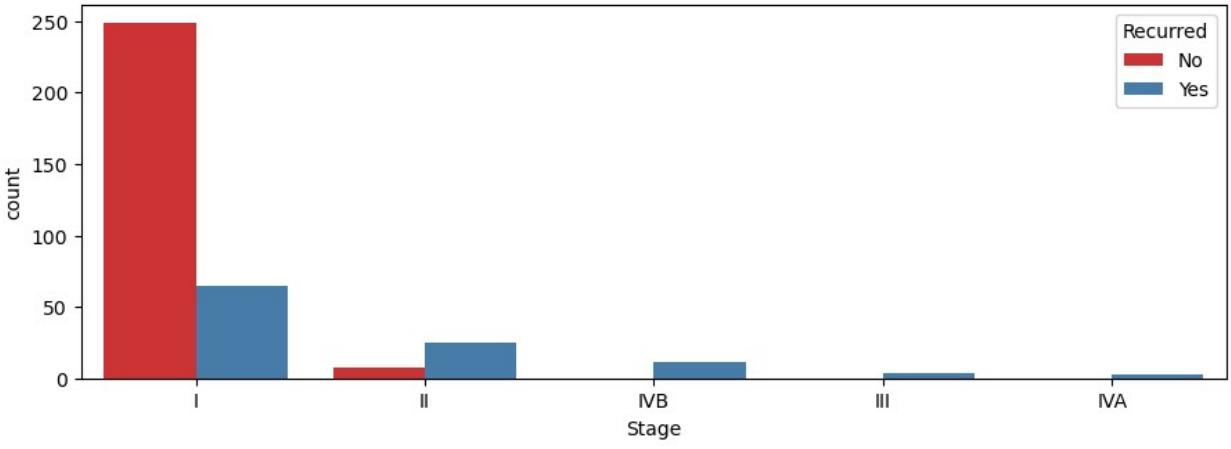
Stage Analysis

```
sns.set_palette('Set1')
fig, ax = plt.subplots(3, 1, figsize=(9, 10))
fig.suptitle('Stage Analysis', fontsize=20, fontweight='bold')
plt.tight_layout()

count = df['Stage'].value_counts()

labels = df['Stage'].value_counts().index.tolist()

#Top ax
sns.countplot(y="Stage",data=df, ax=ax[0])
#middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Stage', hue='Recurred', data=df, ax=ax[2])
plt.show()
```

Stage Analysis

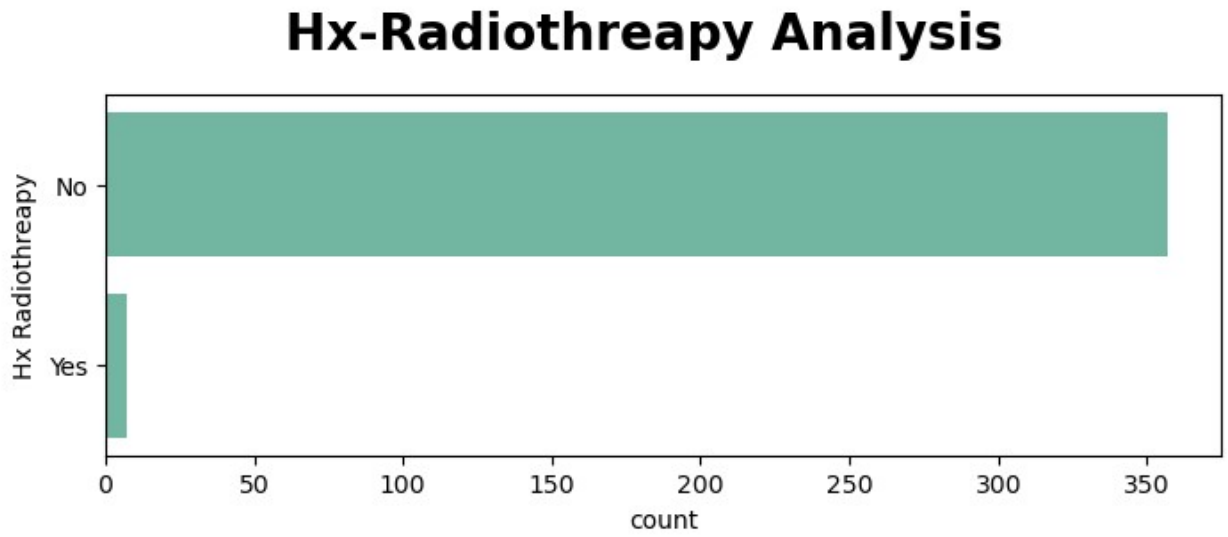


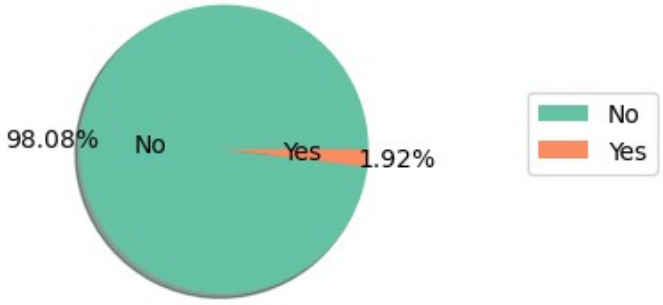


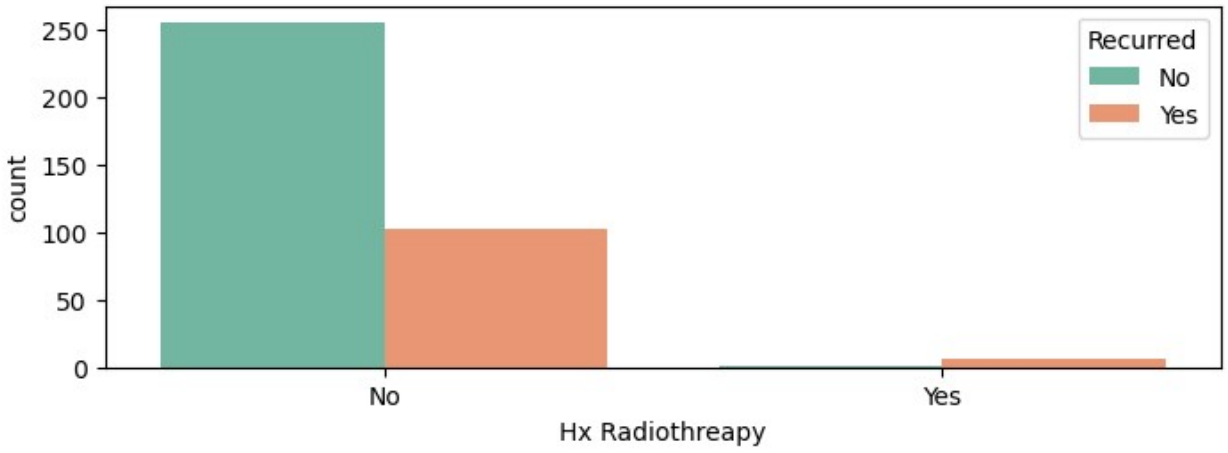


Hx-Radiothreapy Analysis

```
sns.set_palette('Set2')
fig, ax = plt.subplots(3, 1, figsize=(7, 8))
fig.suptitle('Hx-Radiothreapy Analysis', fontsize=20,
fontweight='bold')
plt.tight_layout()

count = df['Hx Radiothreapy'].value_counts()

labels = df['Hx Radiothreapy'].value_counts().index.tolist()

#Top ax
sns.countplot(y="Hx Radiothreapy",data=df, ax=ax[0])
#middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Hx Radiothreapy', hue='Recurred', data=df, ax=ax[2])
plt.show()
```

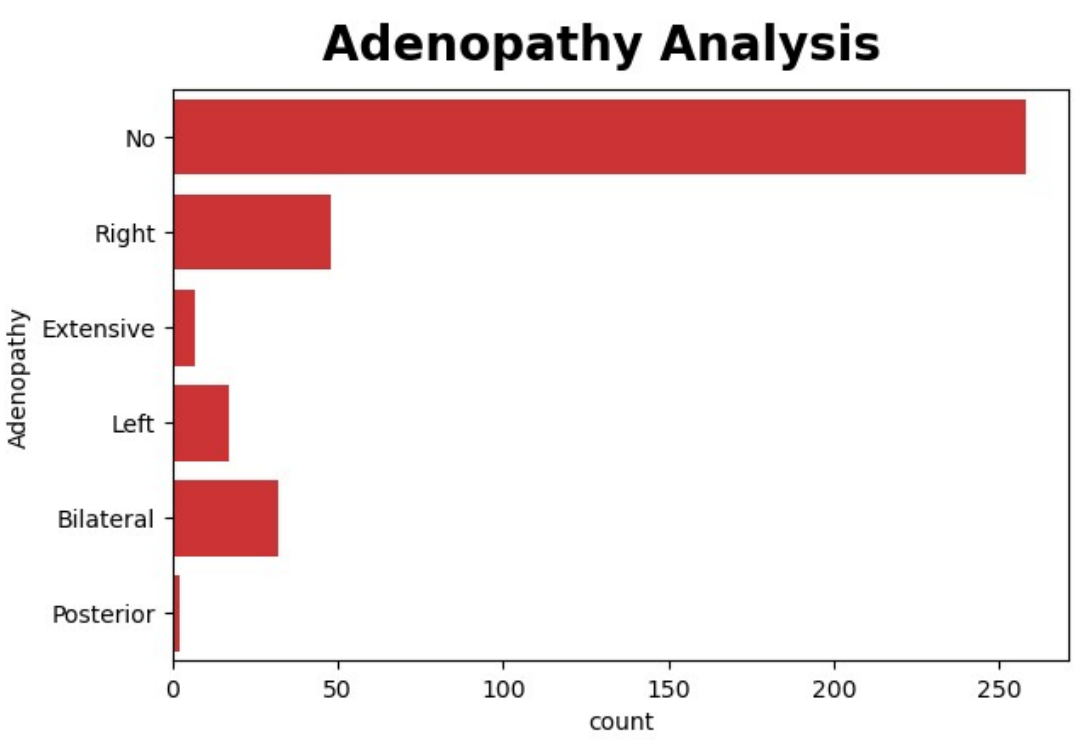


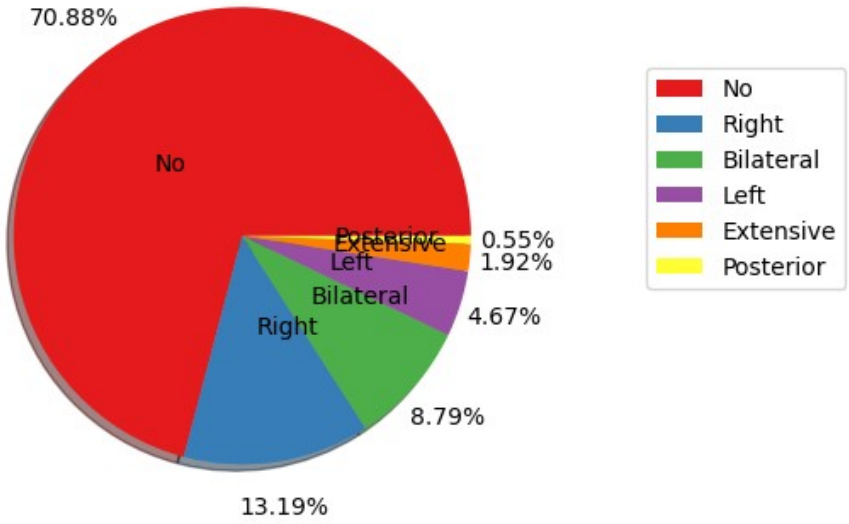




Adenopathy Analysis

```
sns.set_palette('Set1')
fig, ax = plt.subplots(3, 1, figsize=(6, 12))
fig.suptitle('Adenopathy Analysis', fontsize=20, fontweight='bold')
plt.tight_layout()

count = df.Adenopathy.value_counts()

labels = df.Adenopathy.value_counts().index.tolist()

#Top ax
sns.countplot(y="Adenopathy",data=df, ax=ax[0])
#middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Adenopathy', hue='Recurred', data=df, ax=ax[2])
plt.show()
```

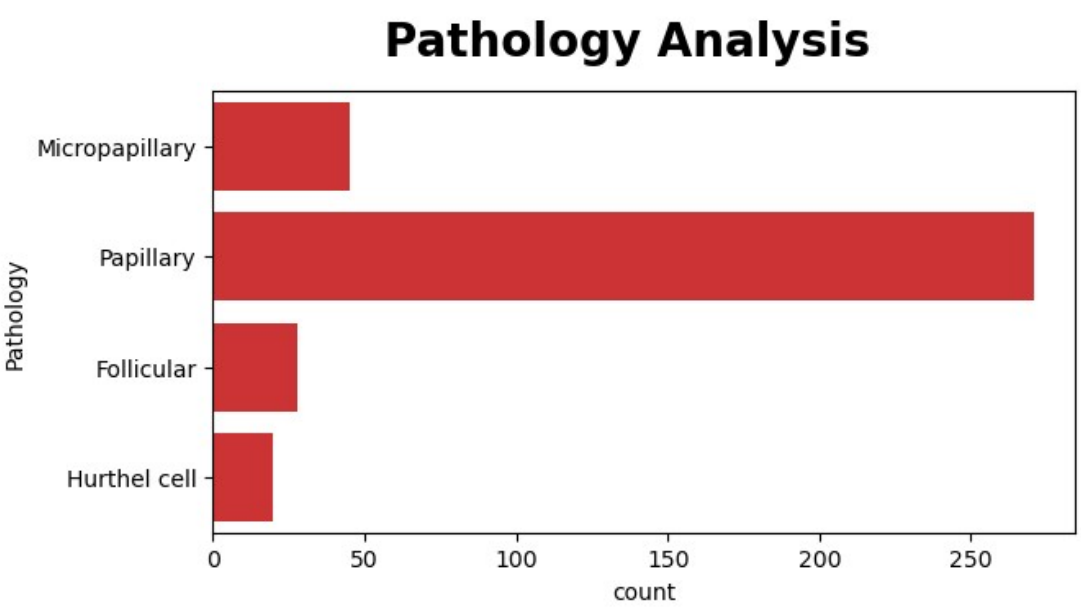

Adenopathy Analysis
No
Right
Extensive
Left
Bilateral
Posterior
Adenopathy
0
50
100
150
200
250
count

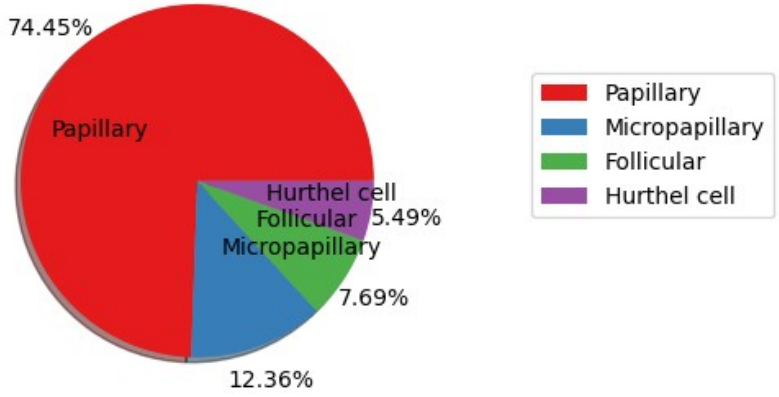

70.88%
No
Posterior
Extensive
Left
Bilateral
Right
0.55%
1.92%
4.67%
8.79%
13.19%
No
Right
Bilateral
Left
Extensive
Posterior

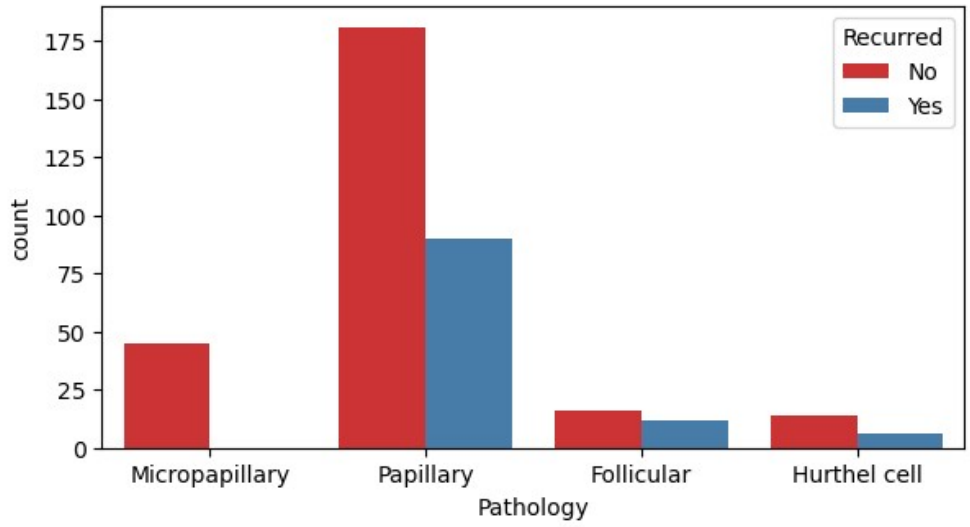

200
150
count
100
Recurred
No
Yes

Pathology Analysis

```
fig, ax = plt.subplots(3, 1, figsize=(6, 10))
fig.suptitle('Pathology Analysis', fontsize=20, fontweight='bold')
plt.tight_layout()

count = df.Pathology.value_counts()

labels = df.Pathology.value_counts().index.tolist()

#Top ax
sns.countplot(y="Pathology",data=df, ax=ax[0])
#middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Pathology', hue='Recurred', data=df, ax=ax[2])
plt.show()
```

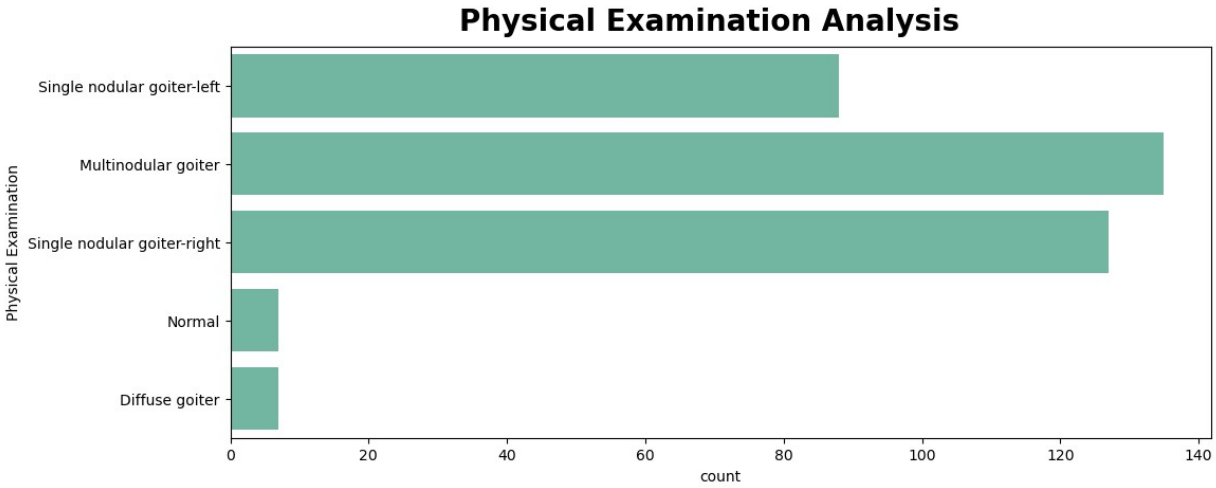

Pathology Analysis
Micropapillary
Papillary
Follicular
Hurthel cell
Pathology
0
50
100
150
200
250
count

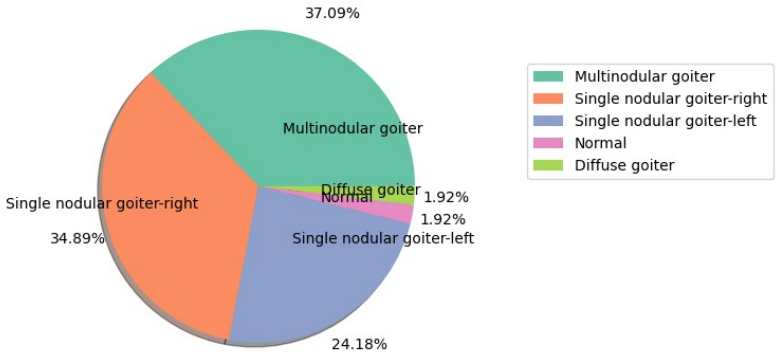

74.45%
Papillary
Hurthel cell
Follicular
5.49%
Micropapillary
7.69%
12.36%
Papillary
Micropapillary
Follicular
Hurthel cell

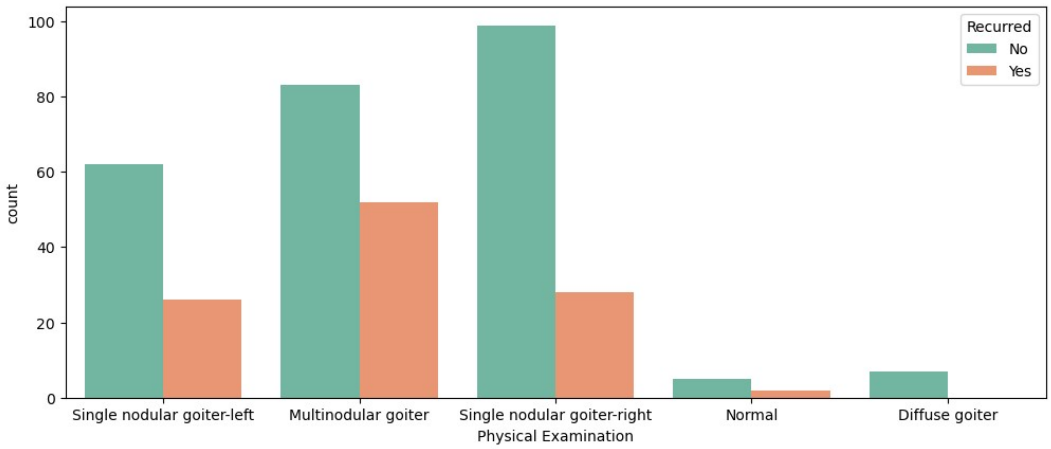

Recurred
No
Yes
175
150
125
100
75
50
25
0
count
Micropapillary
Papillary
Follicular
Hurthel cell
Pathology

Physical Examination Analysis

```
sns.set_palette('Set2')
fig, ax = plt.subplots(3, 1, figsize=(10, 13))
fig.suptitle('Physical Examination Analysis', fontsize=20,
fontweight='bold')
plt.tight_layout()

count = df['Physical Examination'].value_counts()

labels = df['Physical Examination'].value_counts().index.tolist()

#Top ax
sns.countplot(y="Physical Examination",data=df, ax=ax[0])
#Middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Physical Examination', hue='Recurred', data=df,
ax=ax[2])
plt.show()
```



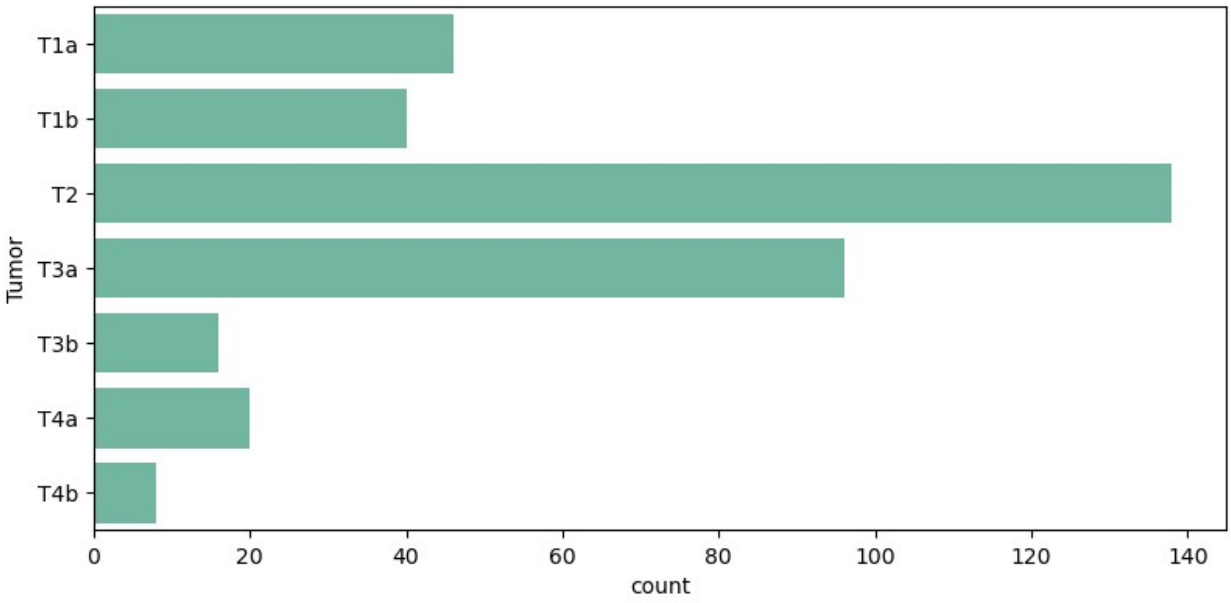


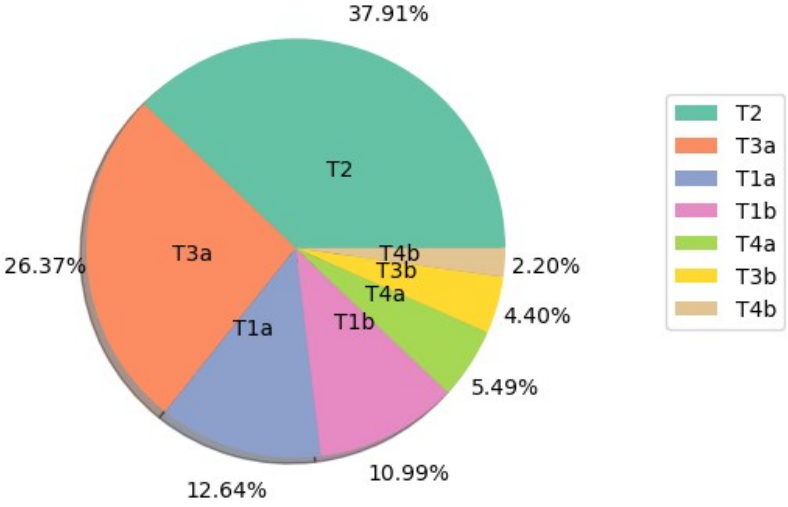


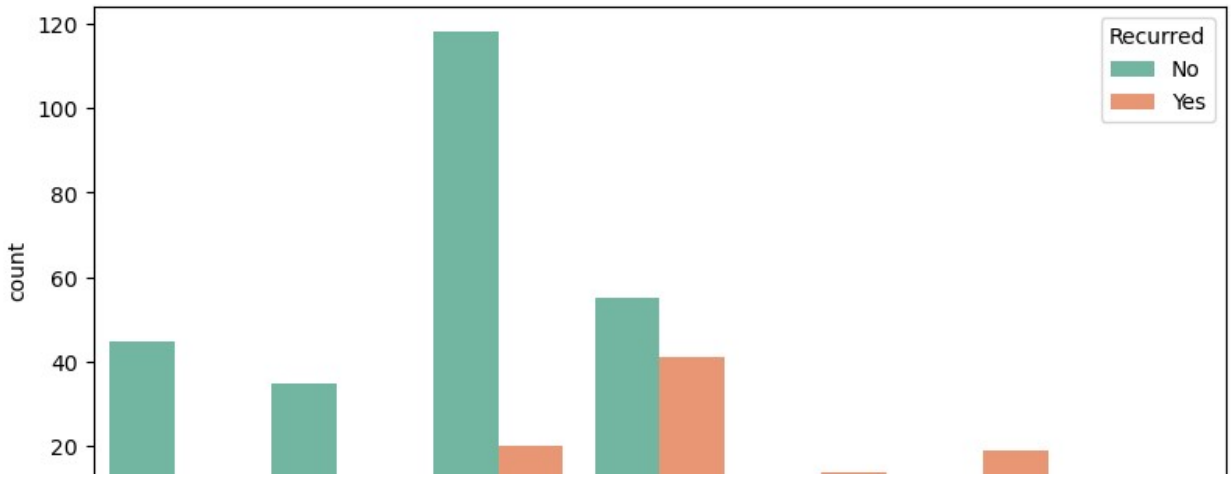
Tumor Analysis

```
sns.set_palette('Set2')
fig, ax = plt.subplots(3, 1, figsize=(8, 12))
fig.suptitle('Tumor Analysis', fontsize=20, fontweight='bold')
plt.tight_layout()

count = df.Tumor.value_counts()

labels = df.Tumor.value_counts().index.tolist()

#Top ax
sns.countplot(y="Tumor",data=df, ax=ax[0])
#middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Tumor', hue='Recurred', data=df, ax=ax[2])
plt.show()
```

Tumor Analysis



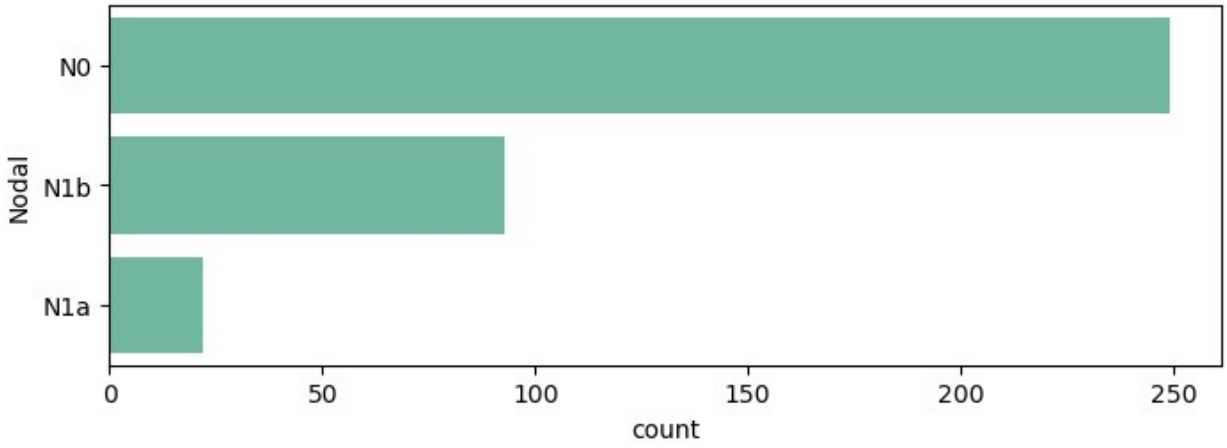


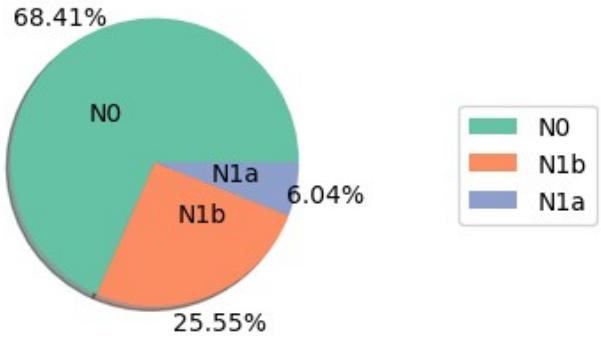


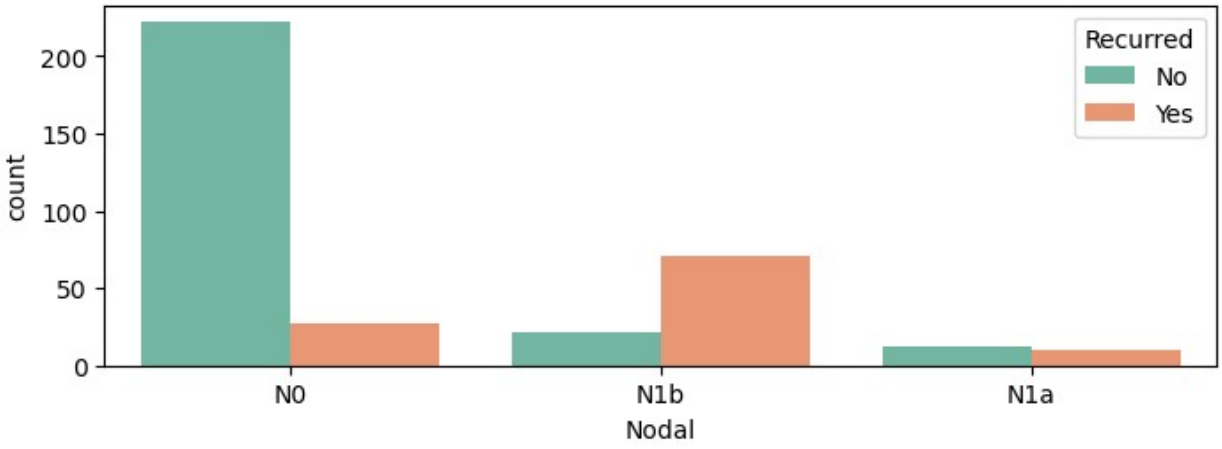
Nodal Analysis

```
sns.set_palette('Set2')
fig, ax = plt.subplots(3, 1, figsize=(7, 8))
fig.suptitle('Nodal Analysis', fontsize=20, fontweight='bold')
plt.tight_layout()

count = df.Nodal.value_counts()

labels = df.Nodal.value_counts().index.tolist()
#Top ax
sns.countplot(y="Nodal",data=df, ax=ax[0])
#Middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Nodal', hue='Recurred', data=df, ax=ax[2])
plt.show()
```

Nodal Analysis

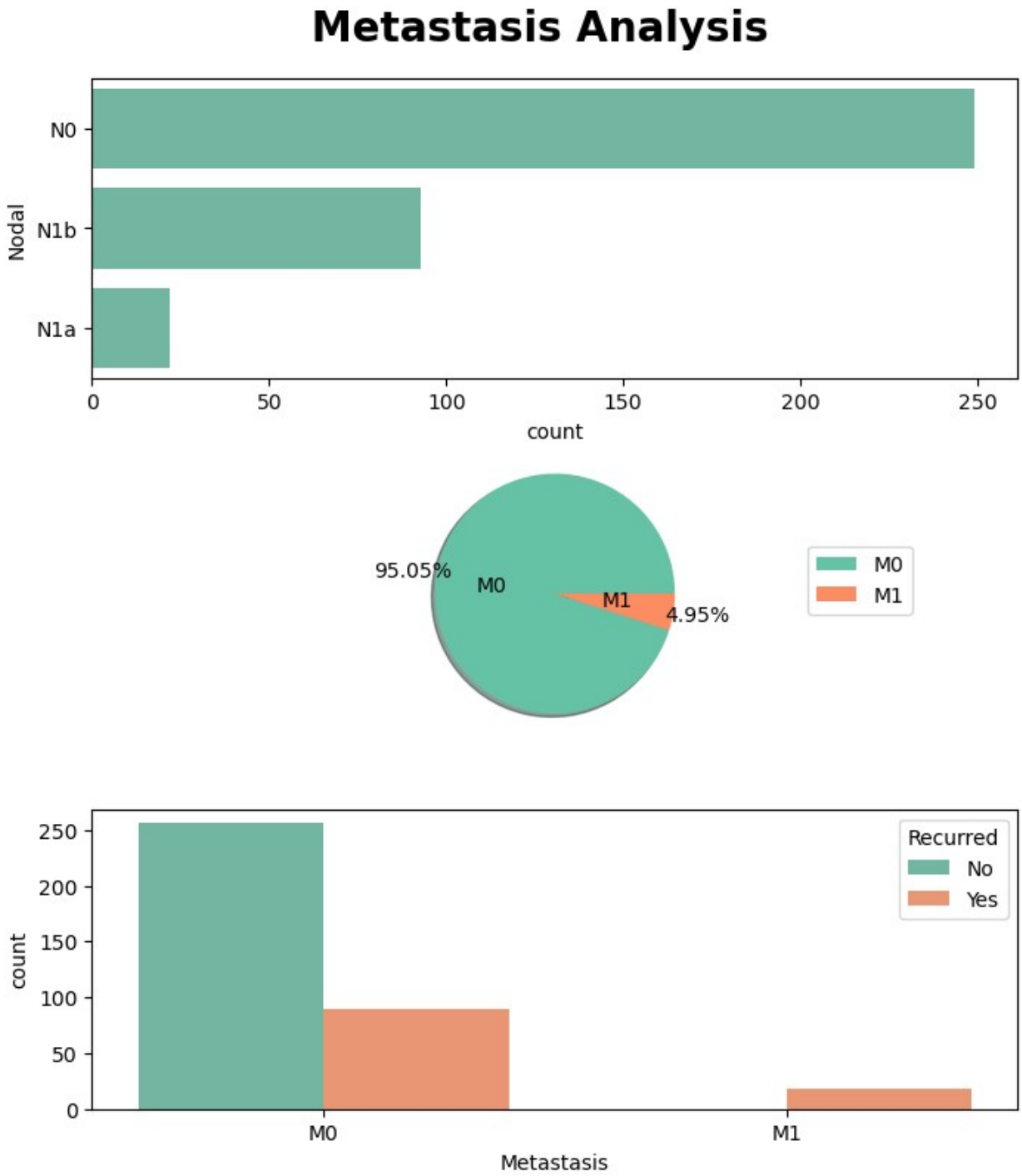






Metastasis Analysis

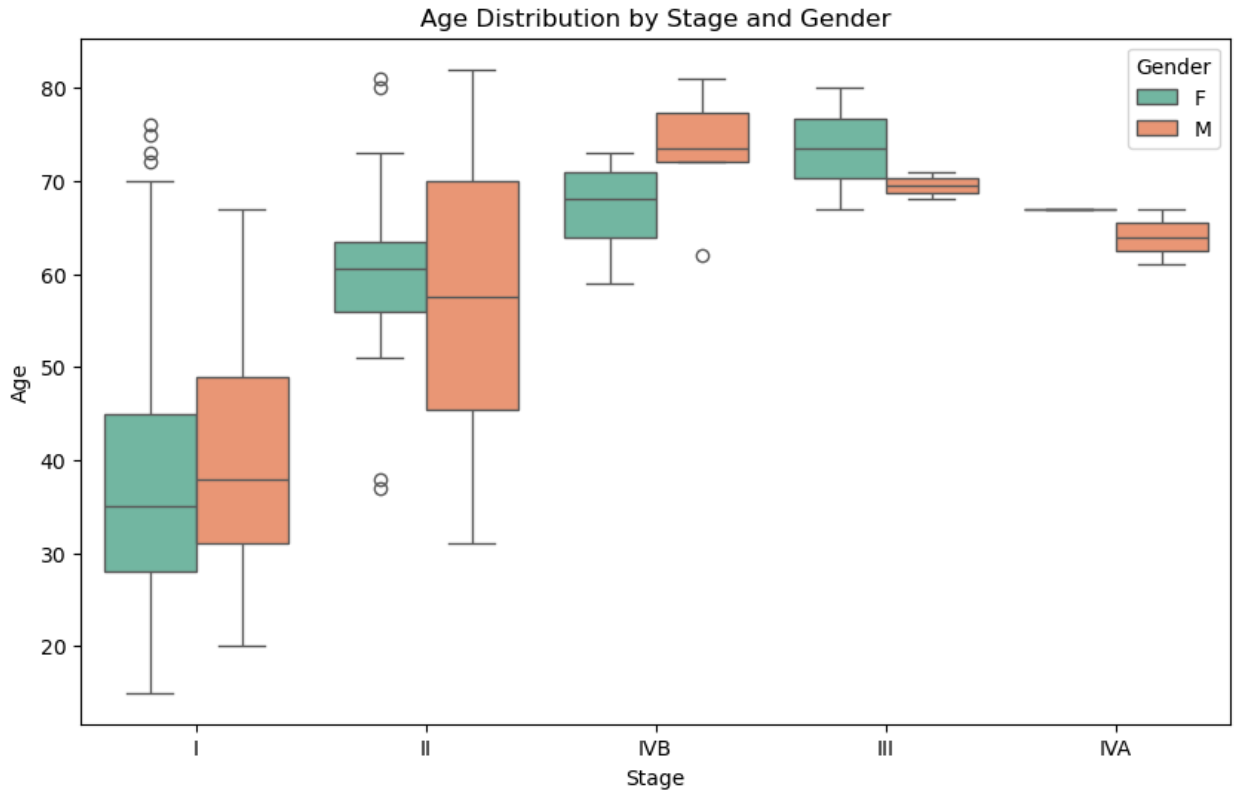
```
sns.set_palette('Set2')
fig, ax = plt.subplots(3, 1, figsize=(7, 8))
fig.suptitle('Metastasis Analysis', fontsize=20, fontweight='bold')
plt.tight_layout()

count = df.Metastasis.value_counts()



labels = df.Metastasis.value_counts().index.tolist()
#Top ax
sns.countplot(y="Nodal",data=df, ax=ax[0])
#Middle ax
ax[1].pie(count, autopct='%.2f%%', labels=labels, shadow=True,
pctdistance=1.2, labeldistance=0.4)
ax[1].legend(bbox_to_anchor=(1, 1), loc=2, borderaxespad=5)

# Bottom ax
sns.countplot(x='Metastasis', hue='Recurred', data=df, ax=ax[2])
plt.show()
```



3.2 Bivariate and Multivariate Analysis

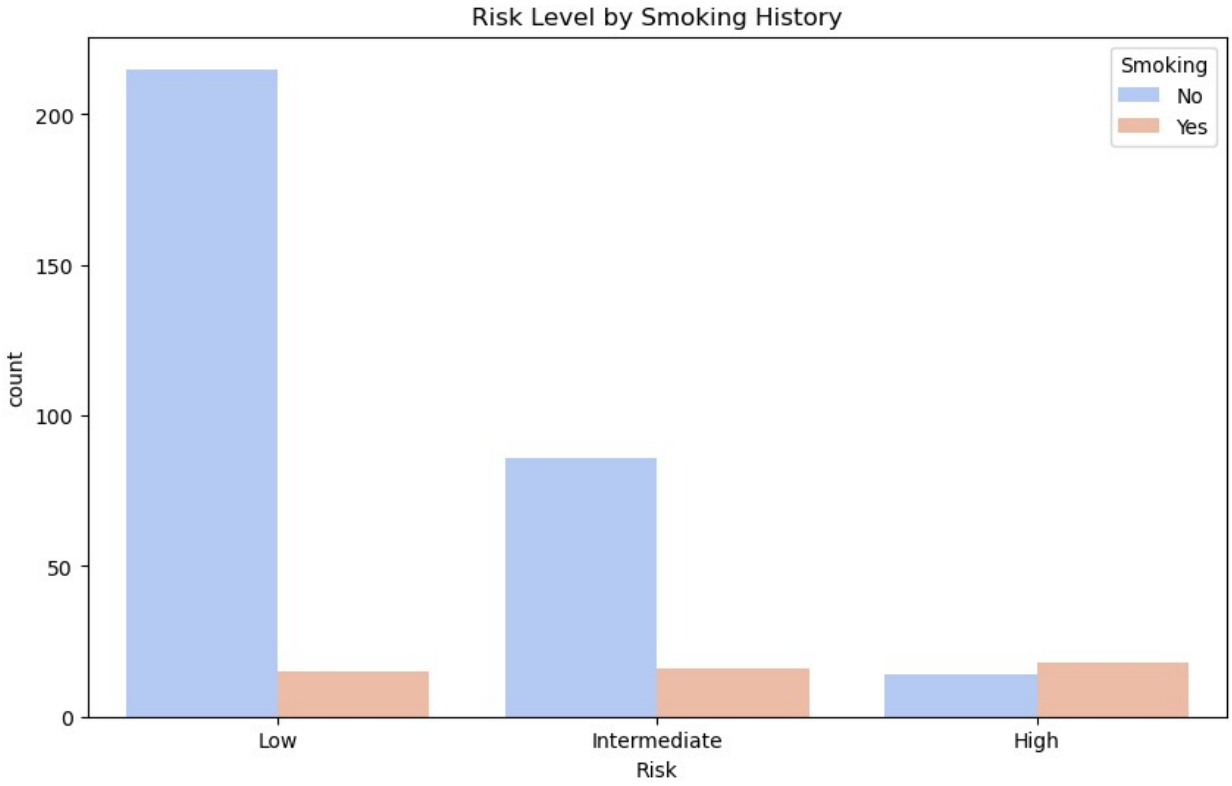
```
# Stage by Age and Gender
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='Stage', y='Age', hue='Gender', palette='Set2')
plt.title('Age Distribution by Stage and Gender')
```

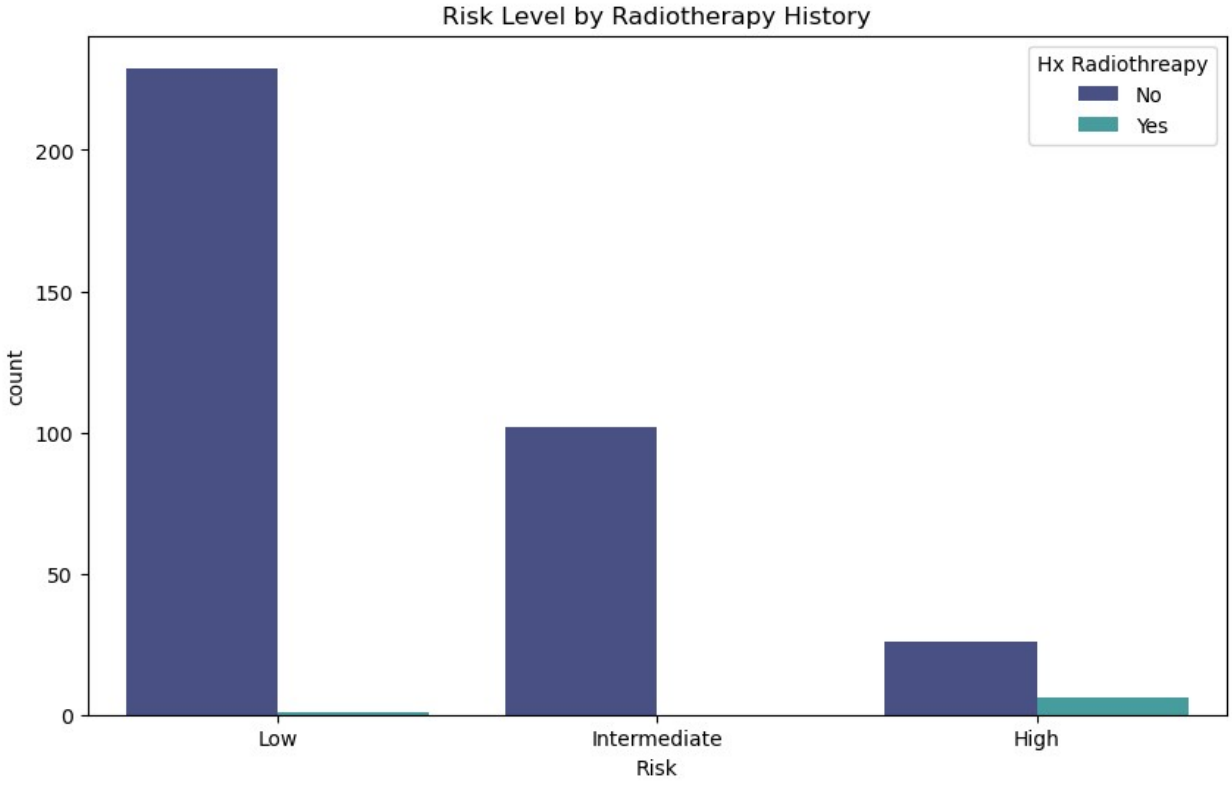
```
Text(0.5, 1.0, 'Age Distribution by Stage and Gender')
```




```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Risk', hue='Smoking', palette='coolwarm')
plt.title('Risk Level by Smoking History')

plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Risk', hue='Hx Radiothreapy',
palette='mako')
plt.title('Risk Level by Radiotherapy History')
```

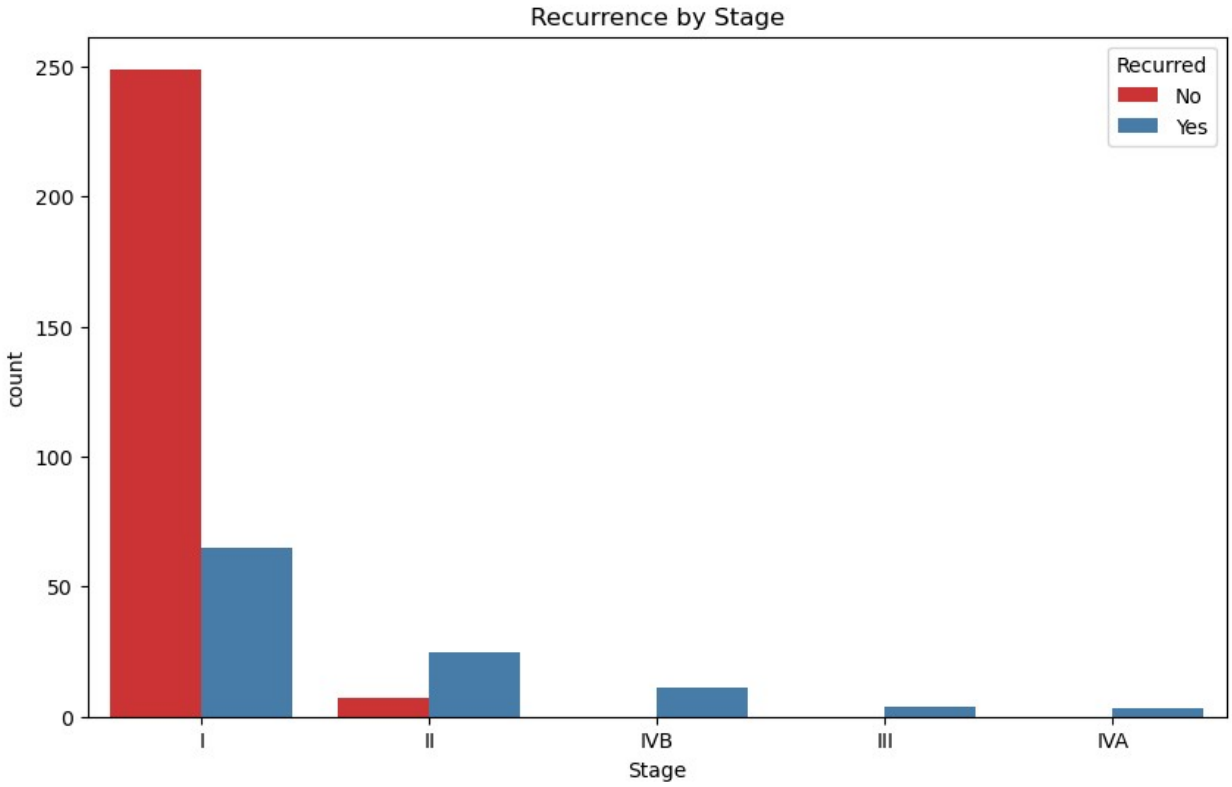
```
Text(0.5, 1.0, 'Risk Level by Radiotherapy History')
```

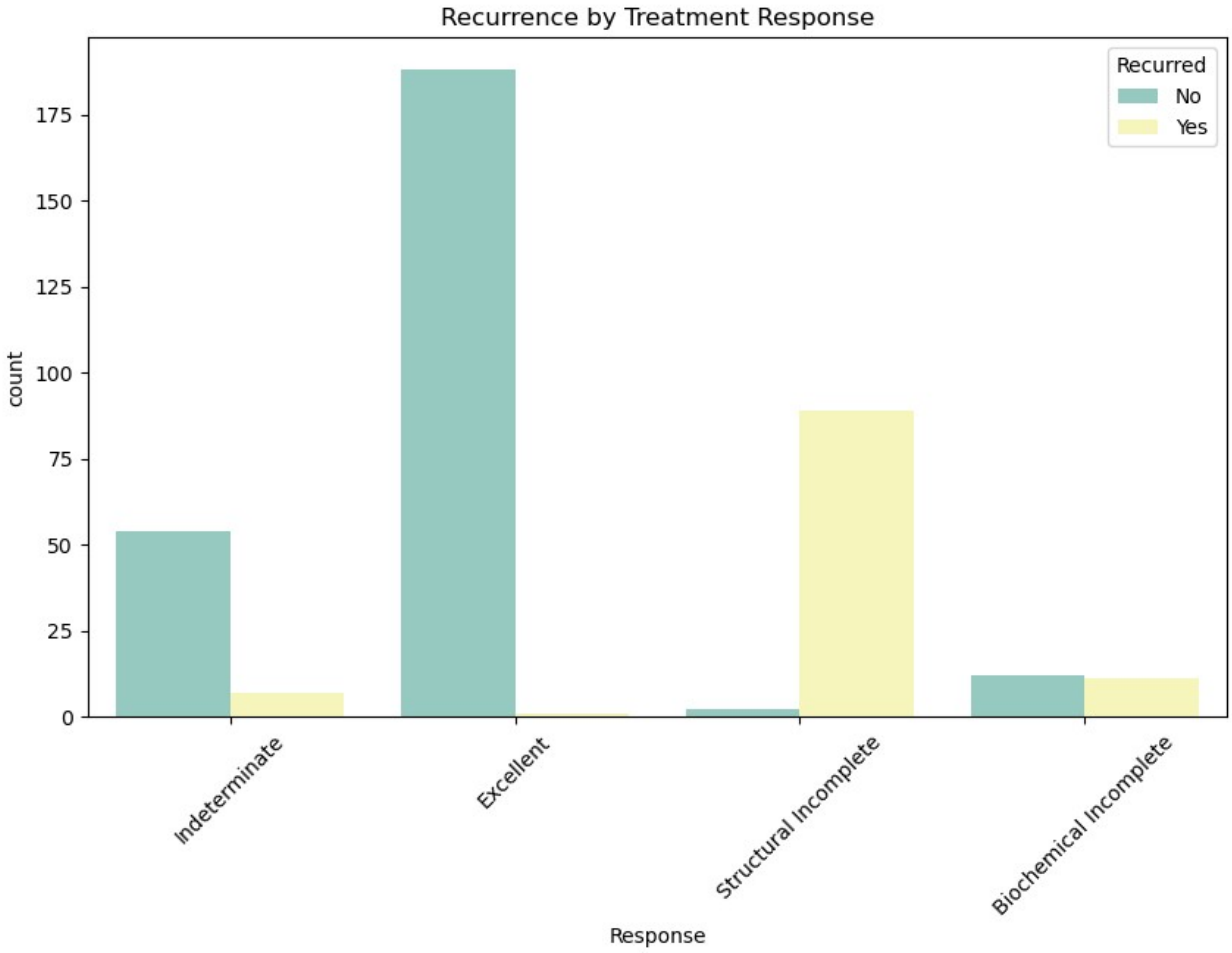

Risk Level by Smoking History
Smoking
No
Yes
count
0
50
100
150
200
Low
Intermediate
High
Risk


Risk Level by Radiotherapy History
Hx Radiothreapy
No
Yes
count
0
50
100
150
200
Low
Intermediate
High
Risk

```
# Recurrence count by Stage
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Stage', hue='Recurred', palette='Set1')
plt.title('Recurrence by Stage')

# Recurrence by Response
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Response', hue='Recurred', palette='Set3')
plt.title('Recurrence by Treatment Response')
plt.xticks(rotation=45)
```

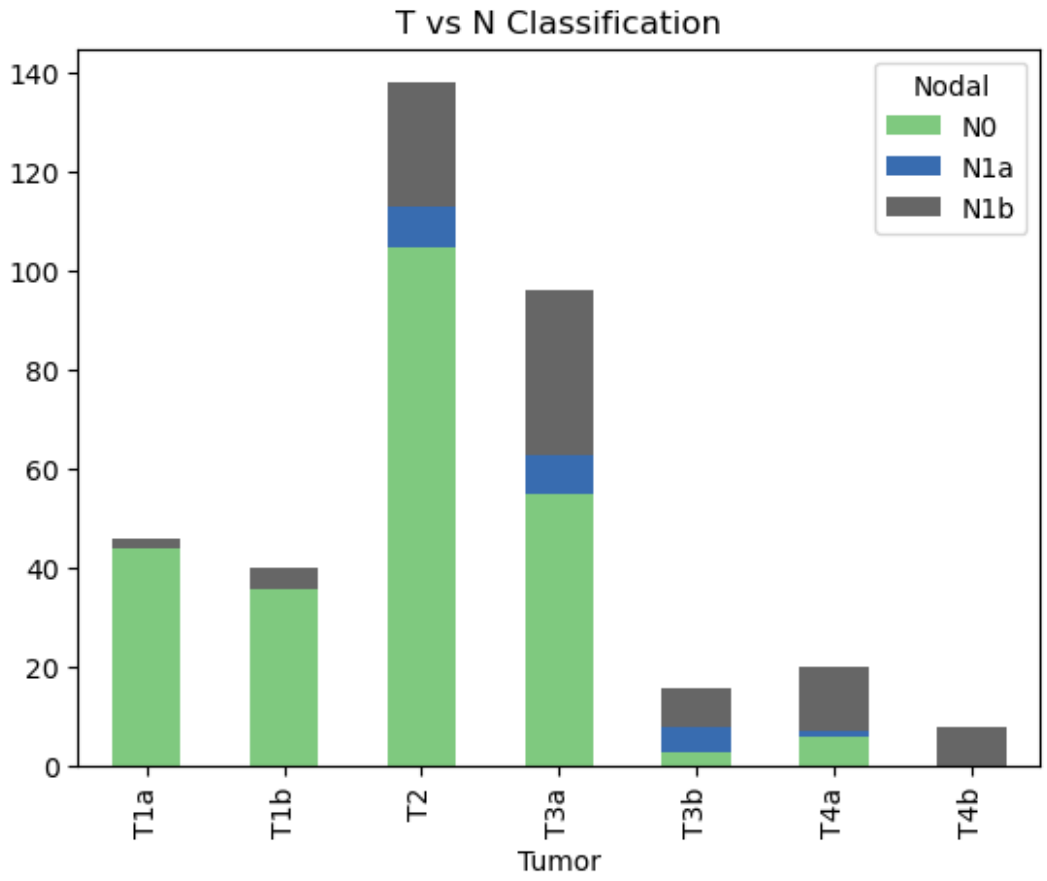
```
([0, 1, 2, 3],
 [Text(0, 0, 'Indeterminate'),
  Text(1, 0, 'Excellent'),
  Text(2, 0, 'Structural Incomplete'),
  Text(3, 0, 'Biochemical Incomplete')])
```

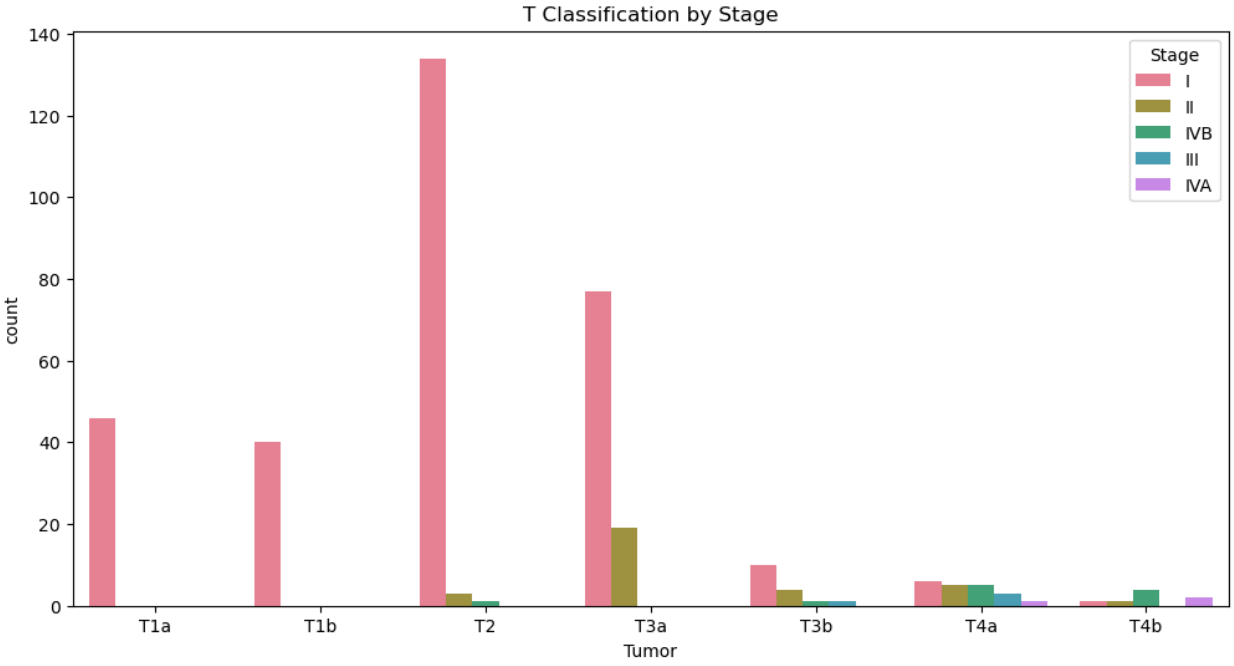





```
# Crosstab heatmap of T and N
pd.crosstab(df['Tumor'], df['Nodal']).plot(kind='bar', stacked=True,
colormap='Accent')
plt.title('T vs N Classification')

# T/N by Stage
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='Tumor', hue='Stage', palette='husl')
plt.title('T Classification by Stage')
```

```
Text(0.5, 1.0, 'T Classification by Stage')
```


T vs N Classification
Nodal
N0
N1a
N1b
140
120
100
80
60
40
20
0
T1a
T1b
T2
T3a
T3b
T4a
T4b
Tumor


T Classification by Stage
Stage
I
II
IVB
III
IVA
count
140
120
100
80
60
40
20
0
T1a
T1b
T2
T3a
T3b
T4a
T4b
Tumor

4. Model Building

Encoding and feature engineering

```
inputs_df = df.drop('Recurred',axis=1)
targets_df= df[['Recurred']]
```

```
categorical_cols =
inputs_df.select_dtypes(include='object').columns.tolist()
categorical_cols
```

```
['Gender',
 'Smoking',
 'Hx Smoking',
 'Hx Radiothreapy',
 'Thyroid Function',
 'Physical Examination',
 'Adenopathy',
 'Pathology',
 'Focality',
 'Risk',
 'Tumor',
 'Nodal',
 'Metastasis',
 'Stage',
 'Response']
```

```
# prepocssesse the categorical
encoder = OneHotEncoder(sparse_output=False)
encoder.fit(inputs_df[categorical_cols])
encoder_cols = encoder.get_feature_names_out(categorical_cols)
inputs_df[encoder_cols]
=encoder.transform(inputs_df[categorical_cols])

final_df =
pd.concat([inputs_df['Age'],inputs_df[encoder_cols]],axis=1)
```

prepocssesse numeric columns

```
scaler = MinMaxScaler()
final_df[['Age']] = scaler.fit_transform(final_df[['Age']])

 X = final_df # independent
 y= targets_df # dependent
```

5.3 Model Training

```
xtrain,xtest,ytrain,ytest =
train_test_split(X,y,test_size=.20,random_state=42)


logist = LogisticRegression()
logist.fit(xtrain,ytrain)
```

```
LogisticRegression()
```

```
y_pred = logist.predict(xtest)

y_pred
```

```
array(['No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No',
'No',
       'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'Yes', 'Yes',
'No',
       'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes',
'No',
       'No', 'Yes', 'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No',
       'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'Yes'], dtype=object)
```

Model Evaluation

```
print(classification_report(ytest,y_pred))
```

```
              precision    recall  f1-score   support

          No       0.96      0.96      0.96        51
         Yes       0.91      0.91      0.91        22
```

```
    accuracy                           0.95        73
   macro avg       0.93      0.93      0.93        73
weighted avg       0.95      0.95      0.95        73
```

```
cm = confusion_matrix(ytest,y_pred)
cm
```

```
array([[49,  2],
       [ 2, 20]], dtype=int64)
```

```
accuracy_score(ytest,y_pred)
```

```
0.9452054794520548
```

5 Advance Modelling

```
random =
RandomForestClassifier(random_state=42,min_samples_split=5,max_depth=16,n_jobs=-1)
random.fit(xtrain,ytrain)
accuracy = accuracy_score(ytest,random.predict(xtest))
print("The accuracy is  : ",accuracy)
```

```
The accuracy is  :  0.9452054794520548
```

```
print(classification_report(ytest,random.predict(xtest)))
```

```
              precision    recall  f1-score   support

          No       0.96      0.96      0.96        51
         Yes       0.91      0.91      0.91        22

    accuracy                           0.95        73
   macro avg       0.93      0.93      0.93        73
weighted avg       0.95      0.95      0.95        73
```

```
cm = confusion_matrix(ytest,random.predict(xtest))
cm
```

```
array([[49,  2],
       [ 2, 20]], dtype=int64)
```