

# CANSER DATASET

## Exploratory Data Analysis (EDA) Project

Duygu Jones | Data Scientist | May 2024

Connect with me: [duygujones.com](https://duygujones.com) | [Linkedin](#) | [GitHub](#) | [Kaggle](#) | [Medium](#) | [Tableau](#)

---

### Table of Contents

1. Introduction
2. Exploratory Data Analysis (EDA)
  - Import The Libraries
  - Read the Dataset
  - Duplicated values
  - Missing values
3. Distributions
  - Distribution of the target variable (malignant vs. benign)
  - Distributions of the key features (size-shape-texture related features)
    - Distributions of the Mean Features
    - Distributions of the Error Features
    - Distributions of the Worst Features
4. Correlations
  - How do the features correlate with each other?
  - What are the correlations between size-related features?
  - Which Features are Positively Correlated?
  - Which Features are Negatively Correlated?
  - Which Features are Uncorrelated?
  - Which features are most correlated in diagnosing malignancy?
5. Summary and Conclusions

### Intoduction

What is the purpose of this exploratory data analysis (EDA) on cancer cell features?

#### Executive Summary:

In this project, performed an Exploratory Data Analysis (EDA) on the **Cancer.csv** dataset using Python.

- **Objective:** To examine the features of **benign** and **malignant** cells and determine the structural similarities and differences between these tumour cells.
- **Dataset:** Each sample belongs to a patient and includes various biopsy measurements that provide information about the characteristics of the tumour cells.








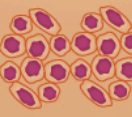
- **Methodology:** Data cleaning, preprocessing, visualization, and analysis to gain insights into the dataset and provide recommendations for further research and potential applications in cancer diagnosis.

## The Problem: Cancer!

Cancer remains one of the most critical health challenges worldwide.

- Cancer is a disease characterized by the uncontrolled growth and spread of abnormal cells in the body (metastasis).
- Cancer can be caused by genetic mutations that can be inherited or acquired due to environmental factors.
- However, it is curable if detected in its early stages as a non-metastatic disease, highlighting the importance of early detection and the need for continued research in this area.

## How does a normal cell differ from a cancer cell?

Normal Cells		Cancer Cells	
Small, uniformly shaped nuclei Relatively large cytoplasmic volume			Large, variable shaped nuclei Relatively small cytoplasmic volume
Conformity in cell size and shape Cells arranged into discrete tissues			Variation in cell size and shape Disorganised arrangement of cells
May possess differentiated cell structures Normal presentation of cell surface markers			Loss of normal specialised features Elevated expression of certain cell markers
Lower levels of dividing cells Cell tissues clearly demarcated			Large number of dividing cells Poorly defined tumor boundaries

DRJOCKERS.COM  
SUPERCHARGE YOUR HEALTH

## For more information:

- [The Difference Between Normal and Cancer Cells](#)
- [Cancer Cells vs Normal Cells](#)

## About the Cancer Dataset

### Structural Analysis

- **Dataset:** Cancer.csv

- **Number of Rows:** 569
- **Number of Columns:** 31

Columns between [0-9] represent the mean\* in the measurements of various tumor characteristics.\*

Columns between [10-19] represent the error rates\* (std) in the measurements of various tumor characteristics.\*

Columns between [20-29] represent the mean of the 3 worst (largest) values\* in the measurements of various tumor characteristics.\*

The column [30] represents: **target:** Type of tumor cell (**1: benign, 0: malignant**).

---

## What are the key features of the cancer dataset?

1. **Radius:** distance from the center to the perimeter.

- *A larger radius indicates a larger tumor, while a smaller radius indicates a smaller tumor.*
- *This measurement can be important in determining the stage and potential malignancy of the tumor, as larger tumors are often more advanced and potentially more dangerous.*

2. **Texture:** standard deviation of gray-scale values (i.e., the pixel intensity). These values are used to quantify the texture.

- *Higher Texture Value: Indicates greater variability in grey-scale intensity, suggesting more heterogeneity and potentially irregular cell structures, which can be associated with malignancy.*
- *Lower Texture Value: Indicates less variability in grey-scale intensity, suggesting smoother and more homogeneous cell surfaces, which is more typical of benign cells.*

3. **Perimeter:**

- *Larger Perimeter: Indicates a larger and potentially more irregular tumor boundary, which may suggest malignancy.(potentially more aggressive)*
- *Smaller Perimeter: Indicates a smaller and potentially more regular tumor boundary, which may suggest benignity.*

4. **Area:** The area can provide important insights into the tumor's growth and stage.

- *Larger Area: Indicates a larger tumor, which may be more advanced or aggressive.*
- *Smaller Area: Indicates a smaller tumor, which may be less advanced.*

5. **Smoothness:** local variation in radius lengths.

- *Higher Smoothness Value: Indicates a more irregular and jagged tumor boundary, suggesting malignancy.*
- *Lower Smoothness Value: Indicates a smoother and more regular tumor boundary, suggesting benignity.*

6. **Compactness:** ( $\text{perimeter}^2 / \text{area} - 1.0$ )

- *Higher Compactness Value: Indicates a more irregular and less compact shape, suggesting malignancy.*
- *Lower Compactness Value: Indicates a more regular and compact shape, suggesting benignity.*

7. **Concavity:** severity of concave portions of the contour.

- *Higher Concavity Value: Indicates more pronounced and numerous inward curvatures in the tumor boundary, suggesting malignancy.*
  - *Lower Concavity Value: Indicates a smoother and more convex boundary, suggesting benignity.*
8. **Concave Points:** number of concave portions of the contour.
- *Higher Number of Concave Points: Indicates a more irregular and complex tumor boundary with more inward curvatures, suggesting malignancy.*
  - *Lower Number of Concave Points: Indicates a smoother and more regular tumor boundary with fewer inward curvatures, suggesting benignity.*
9. **Symmetry:**
- *Higher symmetry values suggest that the cells are more regular and uniform in shape, which is more typical of benign cells.*
  - *Lower symmetry values indicate more irregular, asymmetrical shapes, which can be associated with malignancy.*
10. **Fractal Dimension:** tumor's boundary.
- *Malignant tumors often have more irregular and complex\* boundaries compared to benign.\**

---

### Connections Between the Fields

- **Mean Features:** These features represent the mean values of certain attributes (e.g., radius or area) for each tumor.
  - **Error Features:** These features represent the error rates in the measurements, representing the variability of a particular attribute.
  - **Worst Features:** These features represent the **mean of the 3 worst (largest) measurement of values** and are used to determine the **most aggressive size** or **shape of the tumor**.
- 

## Exploratory Data Analysis (EDA)

### Import The Libraries

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import itertools
from itertools import chain
from sklearn.feature_selection import RFE
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import VotingClassifier
from sklearn.model_selection import GridSearchCV, cross_val_score, learning_curve, train_test_split
from sklearn.metrics import precision_score, recall_score, confusion_matrix, roc_curve, precision_recall_curve

import plotly.offline as py
```

```

py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import plotly.figure_factory as ff

import warnings
warnings.filterwarnings('ignore') #ignore warning messages

```

C:\Users\Duygu Jones\anaconda3\Lib\site-packages\paramiko\transport.py:219: CryptographyDeprecationWarning:

Blowfish has been deprecated and will be removed in a future release

## Read the Dataset

```

In [5]: #Read the "kyphosis.csv" file using Pandas
df0 = pd.read_csv('cancer.csv')
df = df0.copy()

```

## Duplicated values

```

In [6]: #Check the duplicated data if exists
df.duplicated().sum()

```

Out[6]: 0

## Missing values

```

In [7]: df.isnull().sum().sum()

```

Out[7]: 0

## Head and tail

```

In [8]: df.head()

```

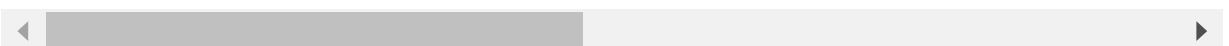
```

Out[8]:

```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	n fr dimer
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.0
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.0
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.0
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.0
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.0

5 rows × 31 columns



```

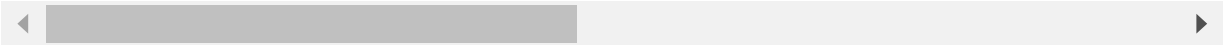
In [9]: df.tail()

```

Out[9]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	dir
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590	
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	
568	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587	

5 rows × 31 columns



General Info

```
In [10]: # Obtain a Statistical Summary about the data
df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   mean radius                           569 non-null    float64
1   mean texture                           569 non-null    float64
2   mean perimeter                         569 non-null    float64
3   mean area                             569 non-null    float64
4   mean smoothness                       569 non-null    float64
5   mean compactness                      569 non-null    float64
6   mean concavity                         569 non-null    float64
7   mean concave points                   569 non-null    float64
8   mean symmetry                         569 non-null    float64
9   mean fractal dimension                569 non-null    float64
10  radius error                           569 non-null    float64
11  texture error                          569 non-null    float64
12  perimeter error                       569 non-null    float64
13  area error                            569 non-null    float64
14  smoothness error                      569 non-null    float64
15  compactness error                     569 non-null    float64
16  concavity error                       569 non-null    float64
17  concave points error                  569 non-null    float64
18  symmetry error                        569 non-null    float64
19  fractal dimension error               569 non-null    float64
20  worst radius                          569 non-null    float64
21  worst texture                         569 non-null    float64
22  worst perimeter                       569 non-null    float64
23  worst area                            569 non-null    float64
24  worst smoothness                      569 non-null    float64
25  worst compactness                     569 non-null    float64
26  worst concavity                       569 non-null    float64
27  worst concave points                  569 non-null    float64
28  worst symmetry                        569 non-null    float64
29  worst fractal dimension               569 non-null    float64
30  target                               569 non-null    int64
dtypes: float64(30), int64(1)
memory usage: 137.9 KB

```

## Calculate the basic statistics

In [13]: *#Check out the basic statistics of the dataframe*

```
df.describe().T
```

Out[13]:

	count	mean	std	min	25%	50%	75%	max
<b>mean radius</b>	569.0	14.127292	3.524049	6.981000	11.700000	13.370000	15.780000	28.110000
<b>mean texture</b>	569.0	19.289649	4.301036	9.710000	16.170000	18.840000	21.800000	39.280000
<b>mean perimeter</b>	569.0	91.969033	24.298981	43.790000	75.170000	86.240000	104.100000	188.500000
<b>mean area</b>	569.0	654.889104	351.914129	143.500000	420.300000	551.100000	782.700000	2501.000000
<b>mean smoothness</b>	569.0	0.096360	0.014064	0.052630	0.086370	0.095870	0.105300	0.163400
<b>mean compactness</b>	569.0	0.104341	0.052813	0.019380	0.064920	0.092630	0.130400	0.345600
<b>mean concavity</b>	569.0	0.088799	0.079720	0.000000	0.029560	0.061540	0.130700	0.426900
<b>mean concave points</b>	569.0	0.048919	0.038803	0.000000	0.020310	0.033500	0.074000	0.201300
<b>mean symmetry</b>	569.0	0.181162	0.027414	0.106000	0.161900	0.179200	0.195700	0.304600
<b>mean fractal dimension</b>	569.0	0.062798	0.007060	0.049960	0.057700	0.061540	0.066120	0.097300
<b>radius error</b>	569.0	0.405172	0.277313	0.111500	0.232400	0.324200	0.478900	2.873000
<b>texture error</b>	569.0	1.216853	0.551648	0.360200	0.833900	1.108000	1.474000	4.885000
<b>perimeter error</b>	569.0	2.866059	2.021855	0.757000	1.606000	2.287000	3.357000	21.980000
<b>area error</b>	569.0	40.337079	45.491006	6.802000	17.850000	24.530000	45.190000	542.200000
<b>smoothness error</b>	569.0	0.007041	0.003003	0.001713	0.005169	0.006380	0.008146	0.031000
<b>compactness error</b>	569.0	0.025478	0.017908	0.002252	0.013080	0.020450	0.032450	0.135000
<b>concavity error</b>	569.0	0.031894	0.030186	0.000000	0.015090	0.025890	0.042050	0.396000
<b>concave points error</b>	569.0	0.011796	0.006170	0.000000	0.007638	0.010930	0.014710	0.052000
<b>symmetry error</b>	569.0	0.020542	0.008266	0.007882	0.015160	0.018730	0.023480	0.078000
<b>fractal dimension error</b>	569.0	0.003795	0.002646	0.000895	0.002248	0.003187	0.004558	0.029000
<b>worst radius</b>	569.0	16.269190	4.833242	7.930000	13.010000	14.970000	18.790000	36.040000
<b>worst texture</b>	569.0	25.677223	6.146258	12.020000	21.080000	25.410000	29.720000	49.540000
<b>worst perimeter</b>	569.0	107.261213	33.602542	50.410000	84.110000	97.660000	125.400000	251.200000



	count	mean	std	min	25%	50%	75%	m
<b>worst area</b>	569.0	880.583128	569.356993	185.200000	515.300000	686.500000	1084.000000	4254.000
<b>worst smoothness</b>	569.0	0.132369	0.022832	0.071170	0.116600	0.131300	0.146000	0.222
<b>worst compactness</b>	569.0	0.254265	0.157336	0.027290	0.147200	0.211900	0.339100	1.058
<b>worst concavity</b>	569.0	0.272188	0.208624	0.000000	0.114500	0.226700	0.382900	1.252
<b>worst concave points</b>	569.0	0.114606	0.065732	0.000000	0.064930	0.099930	0.161400	0.291
<b>worst symmetry</b>	569.0	0.290076	0.061867	0.156500	0.250400	0.282200	0.317900	0.663
<b>worst fractal dimension</b>	569.0	0.083946	0.018061	0.055040	0.071460	0.080040	0.092080	0.207
<b>target</b>	569.0	0.627417	0.483918	0.000000	0.000000	1.000000	1.000000	1.000

## Distributions

### Distribution of the target variable (malignant vs. benign)

```
In [11]: m = df[(df['target'] == 0)]
         b = df[(df['target'] == 1)]
```

```
In [12]: m.shape
```

```
Out[12]: (212, 31)
```

```
In [9]: b.shape
```

```
Out[9]: (357, 31)
```

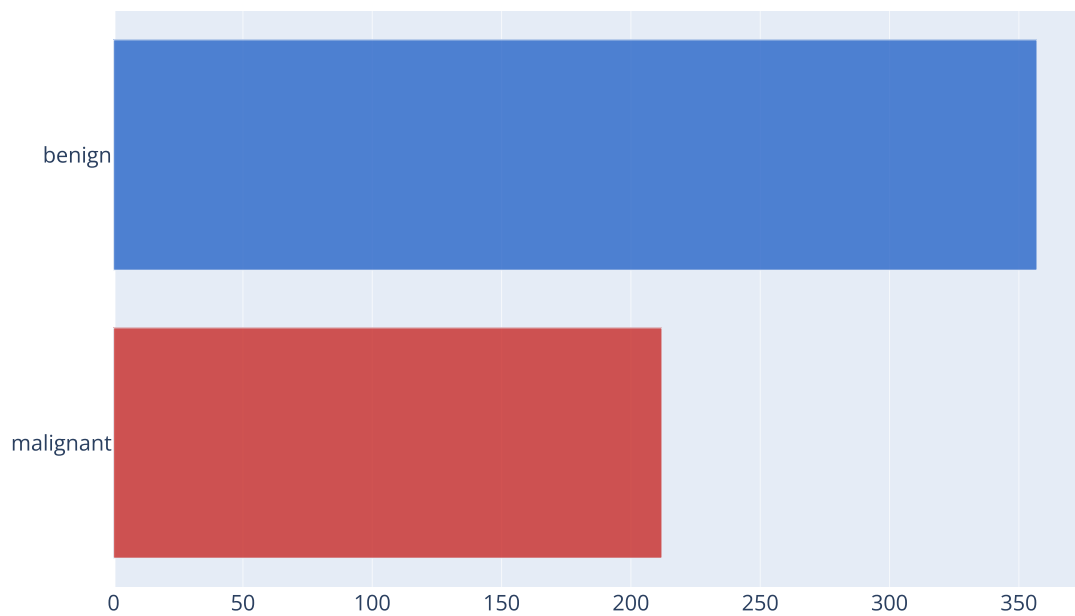
```
In [38]: # COUNT:

plt.figure(figsize=(6,4))
trace = go.Bar(x = (len(m), len(b)), y = ['malignant', 'benign'], orientation = 'h',
               opacity = 0.8, marker=dict(color=['#c82b28', '#2865c8']))

layout = dict(title = 'Count of diagnosis variable', width=700, height=500)

fig = dict(data = [trace], layout=layout)
py.iplot(fig)
```

## Count of diagnosis variable



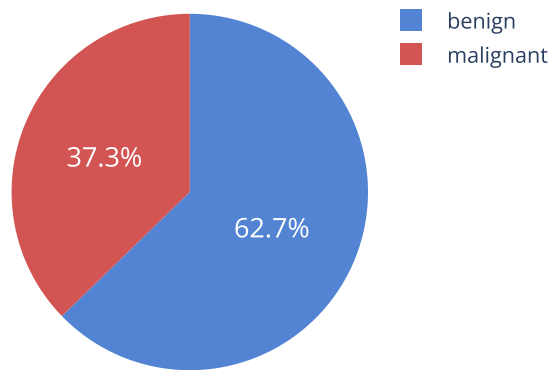
<Figure size 600x400 with 0 Axes>

```
In [34]: #Percentage:
plt.figure(figsize=(5,5))
trace = go.Pie(labels = ['benign','malignant'], values = df['target'].value_counts(),
               textfont=dict(size=15), opacity = 0.8,
               marker=dict(colors=['#2865c8', '#c82b28']))

layout = dict(title='Distribution of diagnosis variable', width=400, height=400)

fig = dict(data = [trace], layout=layout)
py.iplot(fig)
```

## Distribution of diagnosis variable



<Figure size 500x500 with 0 Axes>

💡 Output: Distribution of Target Variable (malignant vs benign)

- The dataset consists of 62.7% benign samples and 37.3% malignant samples.
- There are significantly more benign cases compared to malignant cases.

## What are the distributions of the key features (size-shape-texture related features)?

```
In [41]: def plot_distribution(data_select, size_bin) :
          temporary_m = m[data_select]
          temporary_b = b[data_select]
          hist_data = [temporary_m, temporary_b]

          group_labels = ['malignant', 'benign']
          colors = [ '#c82b28', '#2865c8' ]

          fig = ff.create_distplot(hist_data, group_labels, colors = colors,
                                   show_hist = True, bin_size = size_bin, curve_type='kde')

          fig['layout'].update(title=data_select, width=800, height=500)

          py.iplot(fig, filename = 'Density plot')
```

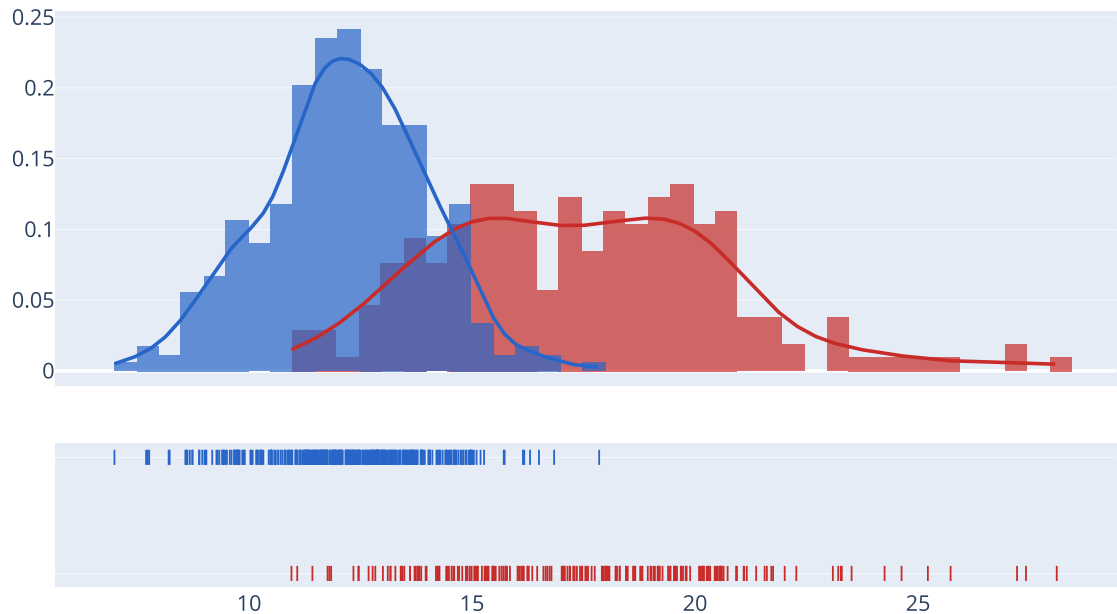
## Distributions of the Mean Features <font

Distributions of Mean Radius by Target:

```
In [42]: plot_distribution('mean radius', .5)
          #plot_distribution('mean texture', .5)
          #plot_distribution('mean perimeter', 5)
          #plot_distribution('mean area', 10)
```

```
#plot_distribution('mean smoothness', .5)
#plot_distribution('mean compactness', .5)
#plot_distribution('mean concavity', .5)
#plot_distribution('mean concave points' .5)
#plot_distribution('mean symmetry', .5)
#plot_distribution('mean fractal dimension', .5)
```

### mean radius



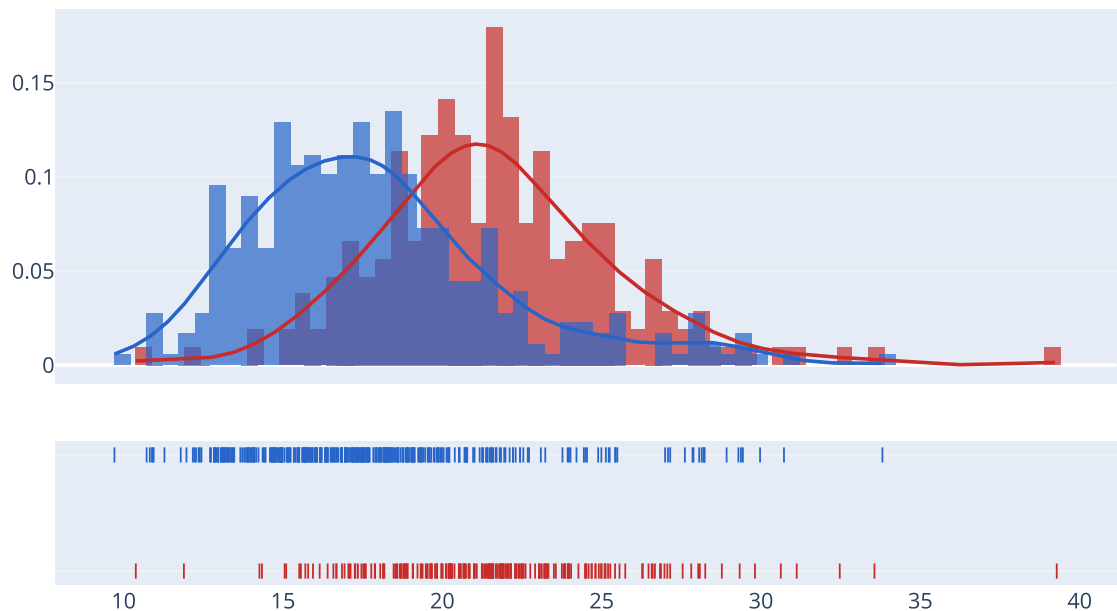
#### 💡 Output: Distribution of Mean Radius by Target

- The histogram and density plot indicate that benign tumors (blue) typically have a lower mean radius, with a peak around 12-14.
- Malignant tumors (red) tend to have a higher and broader range of mean radius values, with many cases above 15.
- There is a clear distinction between benign and malignant tumors based on the mean radius, with malignant tumors generally having larger mean radius values.
  - While `mean radius` can be a useful feature in distinguishing between benign and malignant tumors, there is still overlap that might require additional features for a more accurate classification.

#### Distributions of `Mean texture` by Target:

```
In [43]: plot_distribution('mean texture', .5)
```

## mean texture



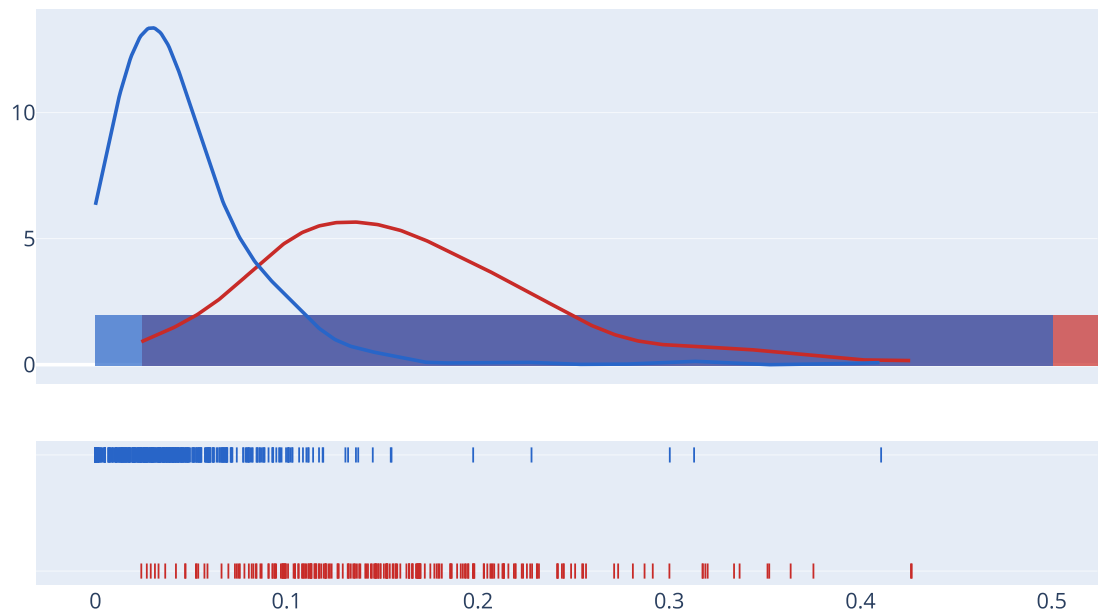
## 💡 Output: Displot; Distribution of Mean Texture

- The distribution of `mean texture` shows that benign tumors (blue) predominantly have values between `15` and `22`.
- Malignant tumors (red), generally have `higher mean texture values`, mostly ranging between `20` and `30`.
- However, there is an `overLap` in the range of `20` to `25`, where both classes share similar values, which might require additional features to fully differentiate between them.

Distributions of `Mean Concavity` by Target:

```
In [44]: plot_distribution('mean concavity', .5)
```

## mean concavity



## 💡 Output: Displot; Distribution of Mean Concavity

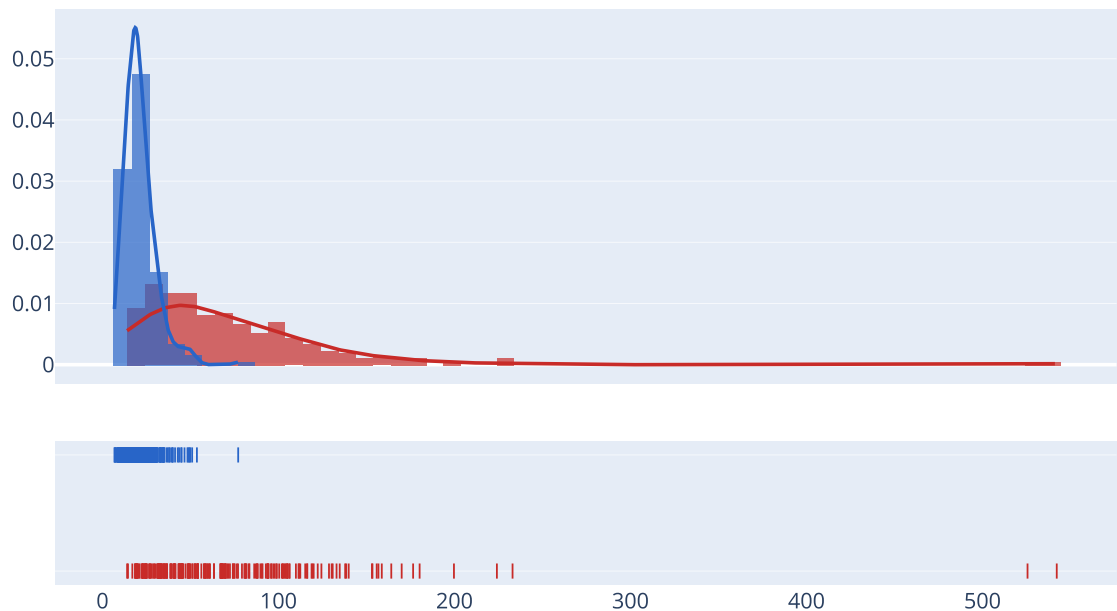
- Benign tumors (blue) predominantly have **low mean concavity values**, whereas malignant tumors (red) display a wider range and generally higher mean concavity values.
- This indicates that mean concavity is a distinguishing feature between benign and malignant tumors, with higher concavity values being more associated with malignancy.

## Distributions of the Error Features

Distributions of Error Area by Target:

```
In [45]: #plot_distribution('radius error', .5)
#plot_distribution('texture error', .5)
#plot_distribution('perimeter error', 5)
plot_distribution('area error', 10)
#plot_distribution('smoothness error', .5)
#plot_distribution('compactness error' .5)
#plot_distribution('concavity error' .5)
#plot_distribution('concave points error' .5)
#plot_distribution('symmetry error' .5)
#plot_distribution('fractal dimension error' .5)
```

## area error



### 💡 Output: Distribution of error area

- The distribution of **error area** shows that benign tumors (blue), **have very low error area values**, mostly concentrated near zero.
- Malignant tumors (red) have a **wider range of error area values** extending beyond 100.

This indicates that **error area is generally higher in malignant tumors compared to benign** ones.

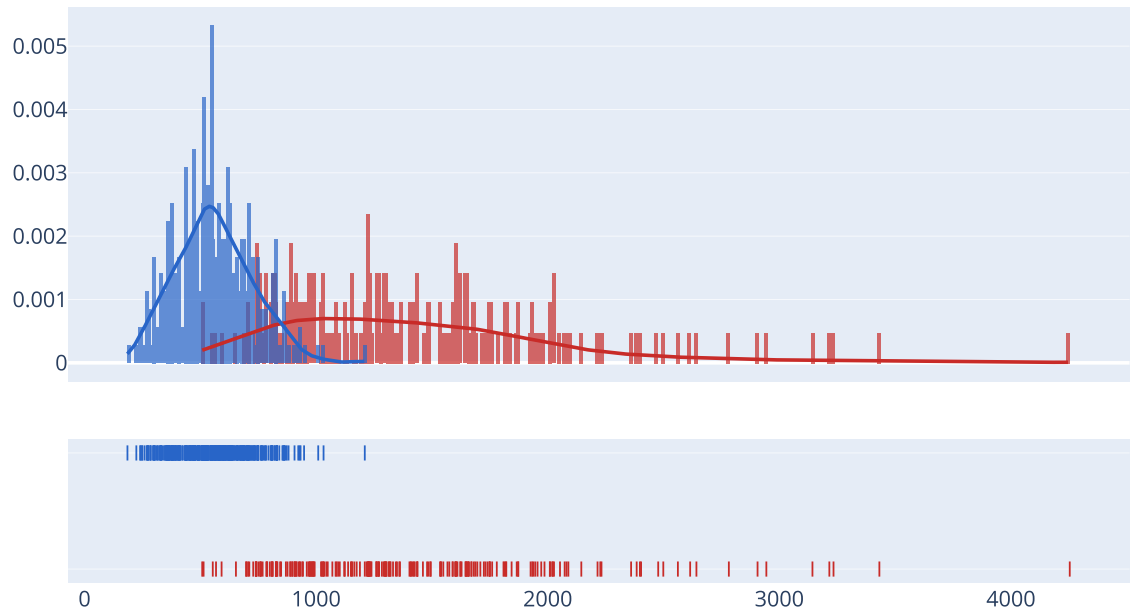
- *However, there is an overlap in the range of 0 to 100, where both classes share similar values, suggesting that additional features might be needed to fully distinguish between them in this range.*

## Distributions of the Worst Features

Distributions of **Worst Area** by Target:

```
In [46]: #plot_distribution('worst radius', .5)
#plot_distribution('worst texture', .5)
#plot_distribution('worst perimeter', 5)
plot_distribution('worst area', 10)
#plot_distribution('worst smoothness', .5)
#plot_distribution('worst compactness' .5)
#plot_distribution('worst concavity' .5)
#plot_distribution('worst concave points' .5)
#plot_distribution('worst symmetry' .5)
#plot_distribution('worst fractal' .5)
```

## worst area



### 💡 Output: Distribution of Worst Area by Target

- The histogram and density plot show that benign tumors (blue) generally have lower "worst area" values, with a peak around 500.
- Malignant tumors (red) tend to have higher "worst area" values, extending well beyond 2000, indicating a broader and more spread out distribution.
- This suggests that "worst area" is a significant feature in distinguishing between benign and malignant tumors, with larger areas typically associated with malignancy.

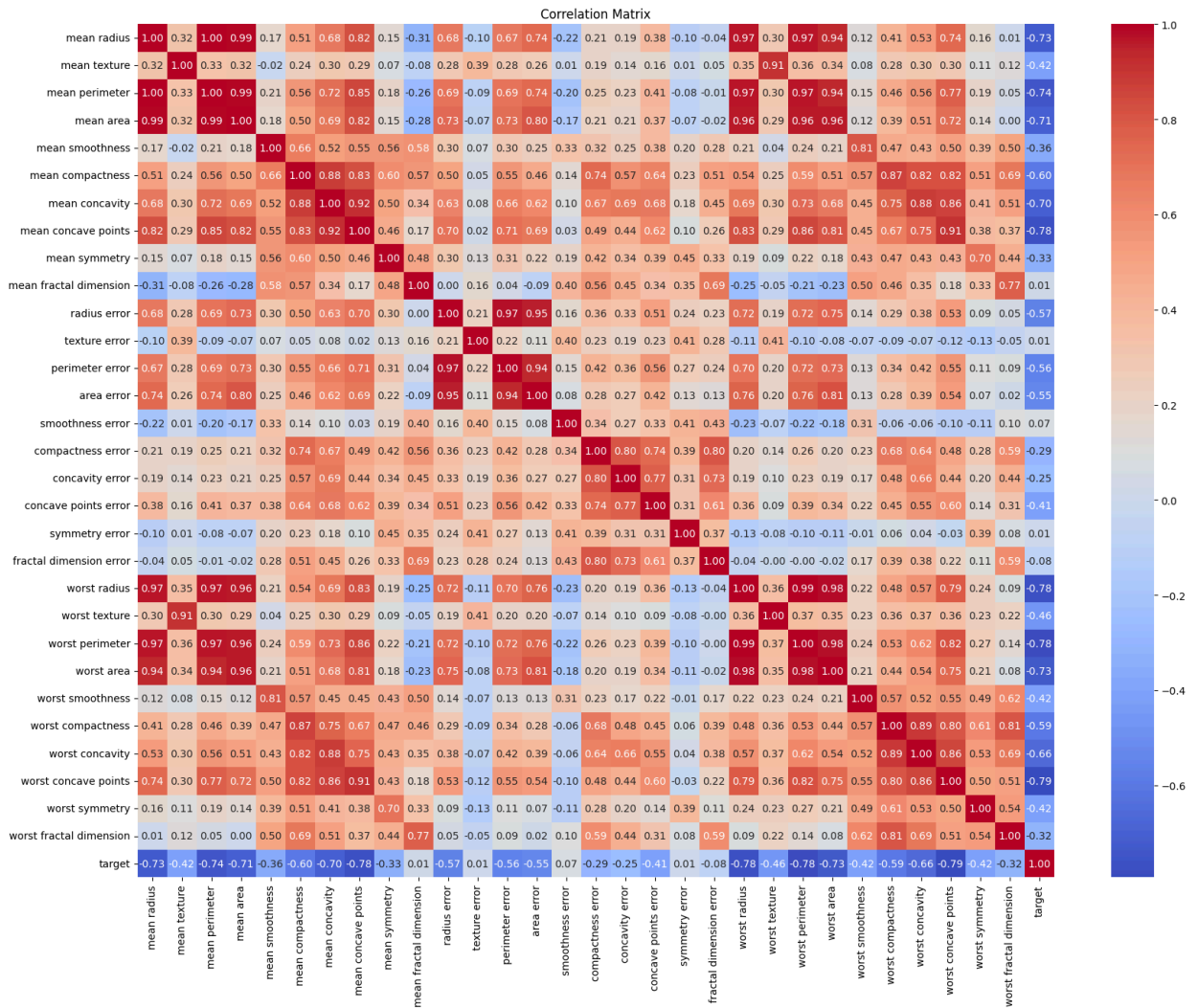
## Correlations

### How do the features correlate with each other?

```
In [22]: # Correlation Matrix

plt.figure(figsize=(20, 15))
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```





### 💡 Output:Correlation Matrix

#### 1. Strong Positive Correlations:

- Features like mean radius, mean area, mean concavity and mean perimeter are highly correlated with each other (around 0.99).

#### 2. Strong Negative Correlations with Target:

- Malignant tumors generally have higher values in radius, texture, and area, but lower values in compactness, smoothness, and symmetry.

These insights highlight key features related to cancer presence and their interrelationships.

What are the correlations between size-related features (area, perimeter, radius)?

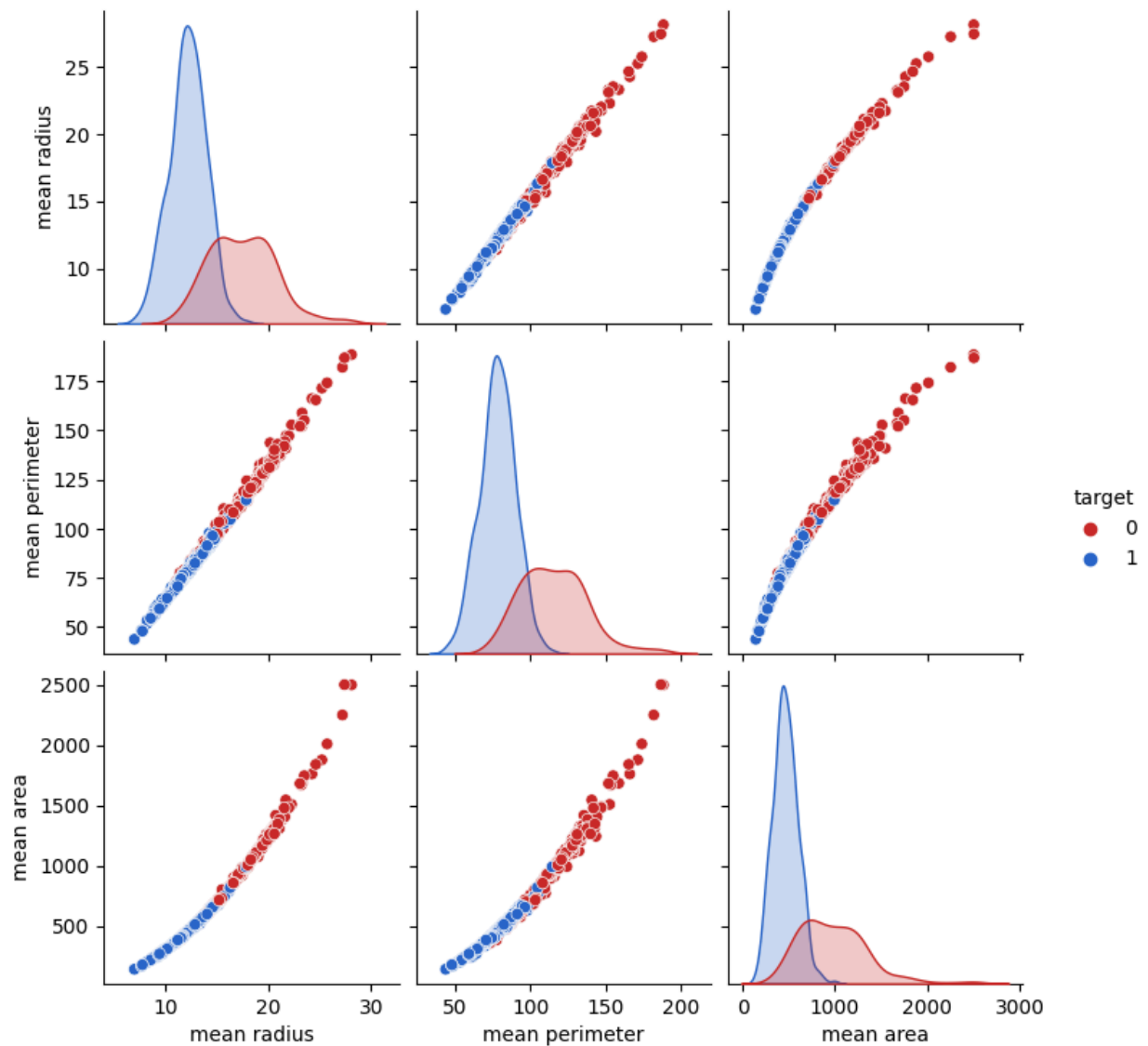
In [23]: # Seaborn: Pairplot of Strong correlation between the mean radius and mean perimeter, mean area

```
sns.pairplot(
    df,
    vars=['mean radius', 'mean perimeter', 'mean area'],
    hue='target',

```

```
palette={
    1: '#2865c8',
    0: '#c82b28'
})

plt.show()
```



💡 Output: correlations between size-related features (area, perimeter, radius)

- **Mean Radius, Mean Perimeter, Mean Area:**

- The strong positive correlations between these features increase together, which is expected as they all relate to the size of the tumor.
- Malignant cases (red) generally have higher values for these features compared to benign cases (blue).

**Overall Insight:**

- **Malignant Cases (Target 0):**

- Tend to have larger values for mean radius, mean perimeter, and mean area.

- **Benign Cases (Target 1):**

- Tend to have smaller and more consistent values for these features.

## Which Features are Positively Correlated?

```
In [24]: palette = {1: '#2865c8', 0: '#c82b28'}
          edgecolor = 'white'

# Plot +
fig = plt.figure(figsize=(10,10))

plt.subplot(221)
ax1 = sns.scatterplot(x = df['mean perimeter'], y = df['worst radius'], hue = "target",
                      data = df, palette = palette, edgecolor=edgecolor)

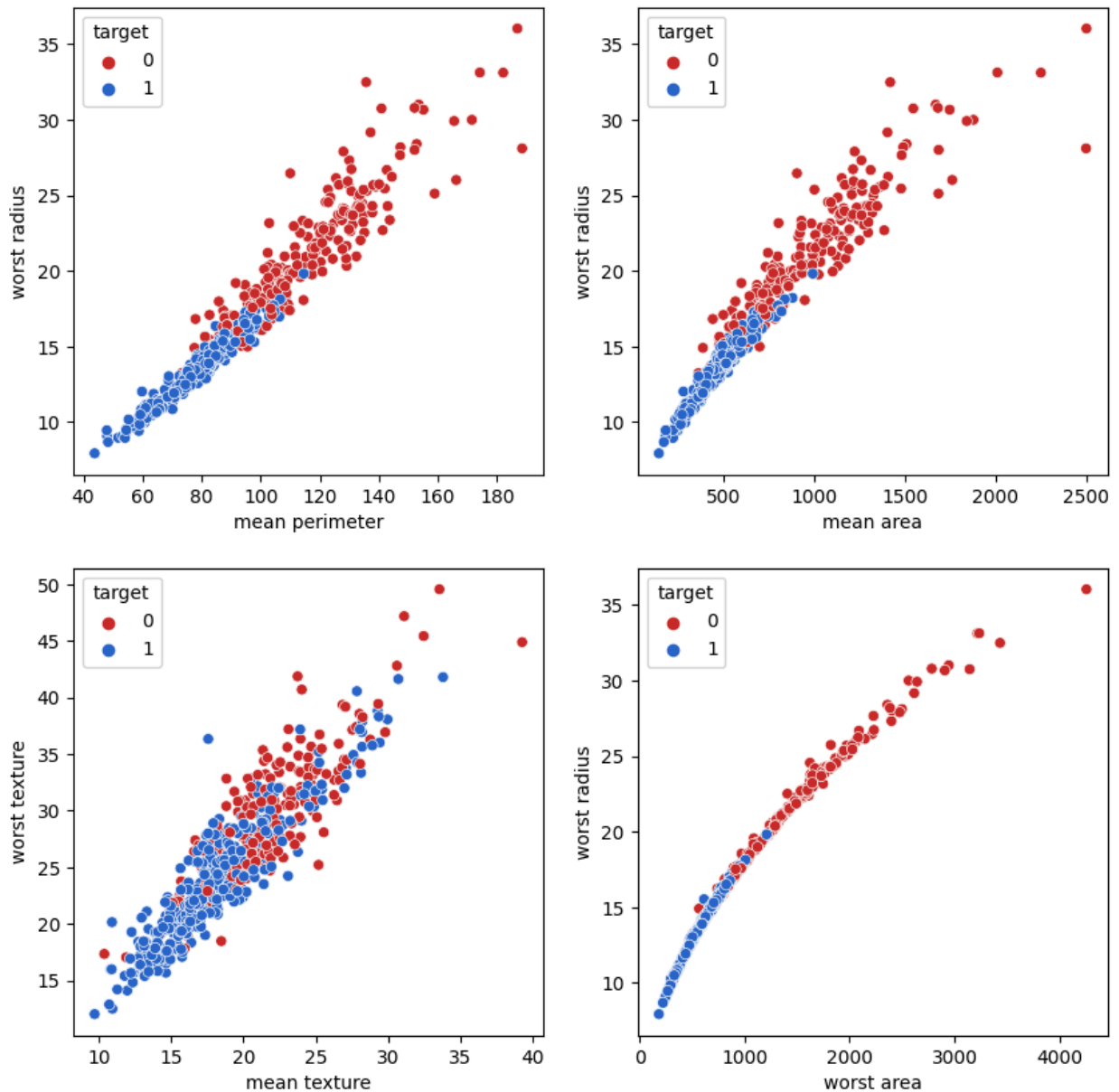
plt.subplot(222)
ax2 = sns.scatterplot(x = df['mean area'], y = df['worst radius'], hue = "target",
                      data = df, palette =palette, edgecolor=edgecolor)

plt.subplot(223)
ax3 = sns.scatterplot(x = df['mean texture'], y = df['worst texture'], hue = "target",
                      data = df, palette =palette, edgecolor=edgecolor)

plt.subplot(224)
ax4 = sns.scatterplot(x = df['worst area'], y = df['worst radius'], hue = "target",
                      data = df, palette =palette, edgecolor=edgecolor)

fig.suptitle('Positive correlated features')
plt.savefig('1')
plt.show()
```

## Positive correlated features



### 💡 Output: Positive Correlated Features vs Target

- All four graphs indicate that higher values in these features (perimeter, area, texture, and radius) are associated with malignant tumors.

**Feature Importance** Size-related features (area, perimeter, radius) and texture irregularities are crucial for distinguishing between malignant and benign tumors.

**Diagnostic Value** These correlations suggest that these features can be used effectively to predict malignancy, aiding in early detection and treatment planning.

## Which Features are Negatively Correlated?

```
In [25]: palette = {1: '#2865c8', 0: '#c82b28'}
edgecolor = 'white'

fig = plt.figure(figsize=(10,10))

plt.subplot(221)
ax1 = sns.scatterplot(x = df['mean radius'], y = df['mean compactness'], hue = "target",
                      data = df, palette =palette, edgecolor=edgecolor)

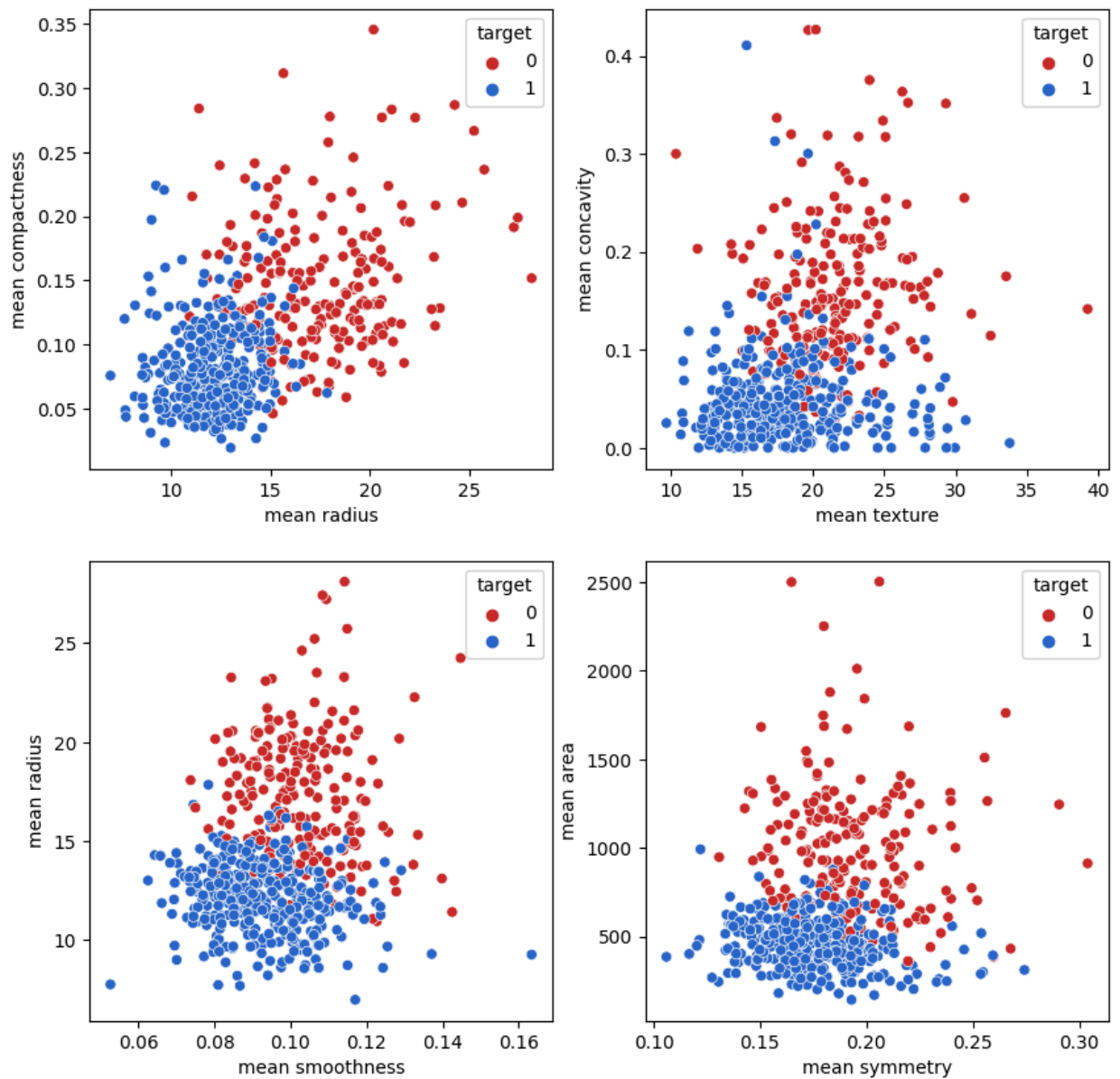
plt.subplot(222)
ax2 = sns.scatterplot(x = df['mean texture'], y = df['mean concavity'], hue = "target",
                      data = df, palette =palette, edgecolor=edgecolor)

plt.subplot(223)
ax2 = sns.scatterplot(x = df['mean smoothness'], y = df['mean radius'], hue = "target",
                      data = df, palette =palette, edgecolor=edgecolor)

plt.subplot(224)
ax2 = sns.scatterplot(x = df['mean symmetry'], y = df['mean area'], hue = "target",
                      data = df, palette =palette, edgecolor=edgecolor)

fig.suptitle('Negative correlated features')
plt.savefig('3')
plt.show()
```

## Negative correlated features



### 💡 Output: Negative Correlation with Target

- These graphs show that as one feature increases, the other tends to decrease, with a clear distinction between malignant (red) and benign (blue) tumors.

**Feature Characteristics:** Malignant tumors generally have higher values in radius, texture, and area, but lower values in compactness, concavity, smoothness, and symmetry.

**Diagnostic Implications:** These relationships suggest that these features can effectively differentiate between malignant and benign tumors, aiding in the diagnostic process.

## Which Features are Not Correlated?

```
In [26]: palette = {1: '#2865c8', 0: '#c82b28'}
         edgecolor = 'white'

fig = plt.figure(figsize=(10,10))

plt.subplot(221)
ax1 = sns.scatterplot(x = df['mean smoothness'], y = df['mean texture'], hue = "target",
                     data = df, palette =palette, edgecolor=edgecolor)

plt.subplot(222)
ax2 = sns.scatterplot(x = df['mean radius'], y = df['mean fractal dimension'], hue = "target",
                     data = df, palette =palette, edgecolor=edgecolor)

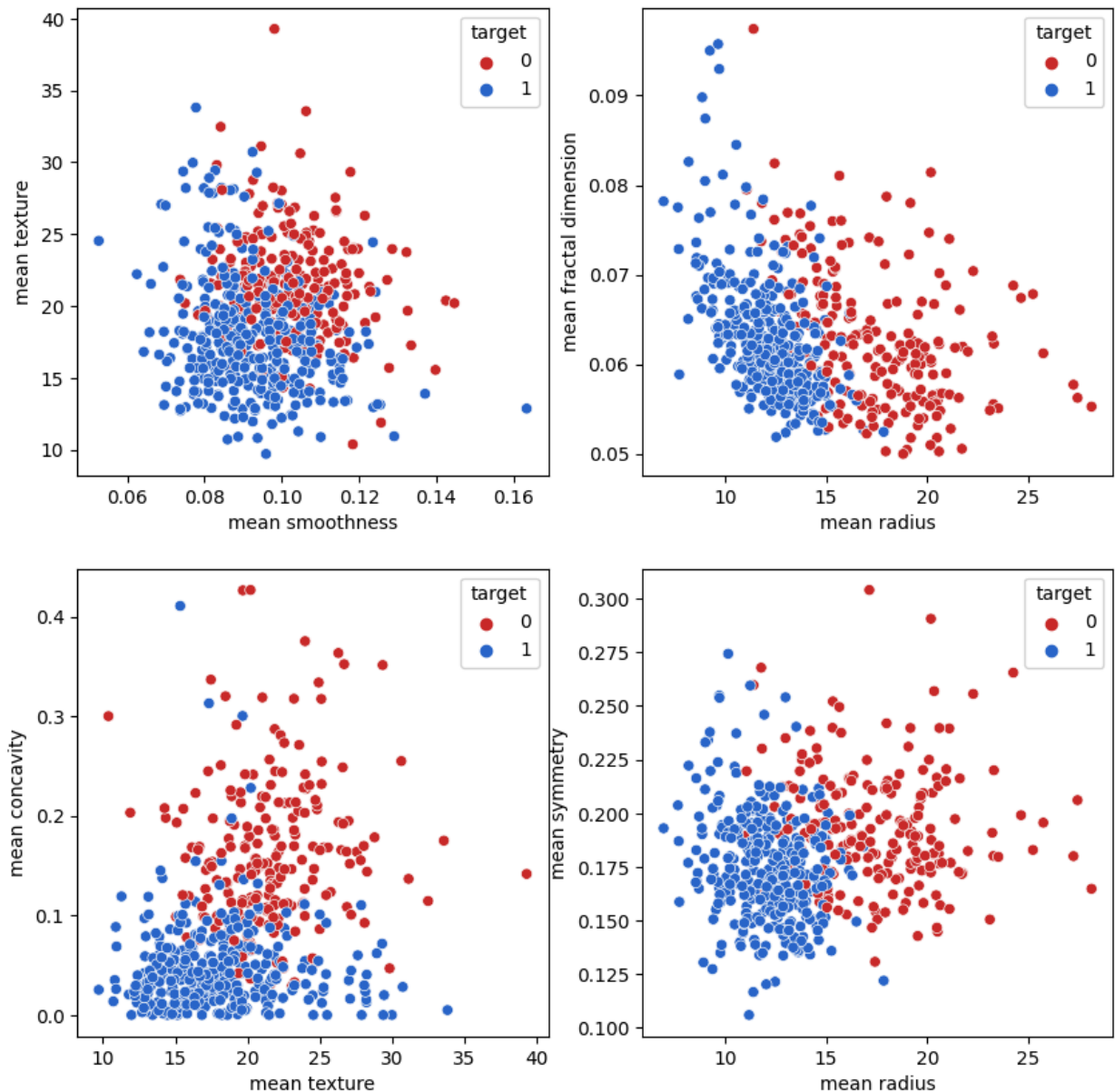
plt.subplot(223)
ax3 = sns.scatterplot(x = df['mean texture'], y = df['mean concavity'], hue = "target",
                     data = df, palette =palette, edgecolor=edgecolor)

plt.subplot(224)
ax4 = sns.scatterplot(x = df['mean radius'], y = df['mean symmetry'], hue = "target",
                     data = df, palette =palette, edgecolor=edgecolor)

legend_labels = ['1: Benign', '0: Malignant']
legend_colors = ['#2865c8', '#c82b28']

fig.suptitle('Uncorrelated features')
plt.savefig('2')
plt.show()
```

### Uncorrelated features



#### 💡 Output: Uncorrelated Features with Target:

- These graphs indicate that the selected features do not have strong relationships with the target variable (malignant vs. benign).

**Diagnostic Implications:** The lack of clear separation between malignant and benign tumors suggests that these features are less useful for diagnostic purposes.

**Overall:** Features such as mean smoothness, mean texture, mean fractal dimension, and mean symmetry may have limited predictive value due to their lack of correlation with the target variable.

### Which features are most correlated in diagnosing malignancy?



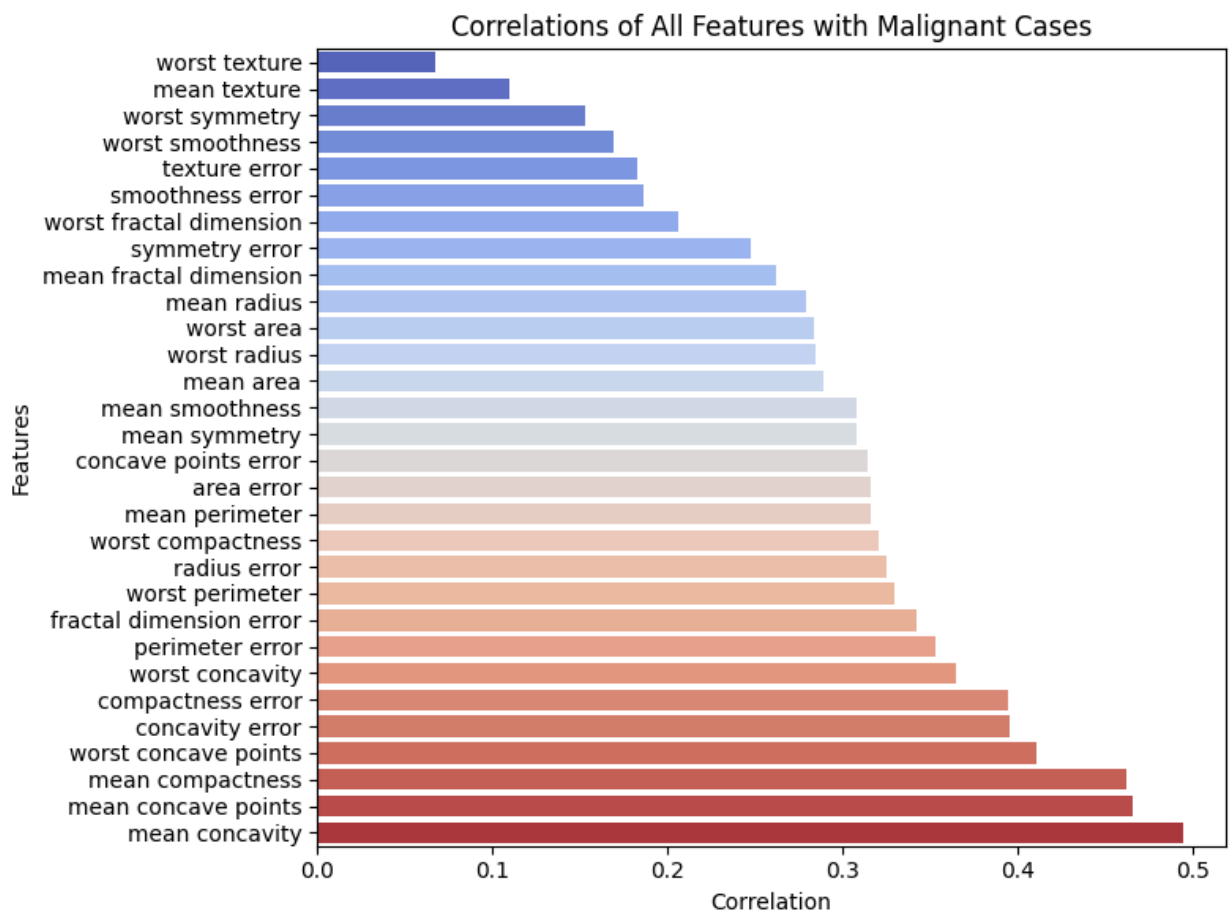
```
In [28]: # Filter the data for target = 0 (malignant)
df_malignant = df[df['target'] == 0]

# Drop the 'target' column
df_malignant = df_malignant.drop(columns=['target'])

# Calculating the correlation matrix for malignant cases
correlation_matrix_malignant = df_malignant.corr()

# Sorting the correlations with the mean correlation of all features
correlations_malignant_sorted = correlation_matrix_malignant.mean().sort_values()

# Plotting for malignant cases
plt.figure(figsize=(8,6))
sns.barplot(y=correlations_malignant_sorted.index, x=correlations_malignant_sorted.values, palette='magma')
plt.title("Correlations of All Features with Malignant Cases")
plt.xlabel("Correlation")
plt.ylabel("Features")
plt.tight_layout()
plt.show()
```



💡 Output: Correlation of all features with the target variable 0: malignancy.

#### 1. Most Correlated Fields:

- Features such as **concavity, compactness perimeter, area**, and **dimension** variables highly positive correlations,
- Indicating that as these features increase, the likelihood of a tumor being malignant also increases.

```
In [29]: # Filter the data for target = 0 (malignant)
df_malignant = df[df['target'] == 0]

# Drop the 'target' column
df_malignant = df_malignant.drop(columns=['target'])

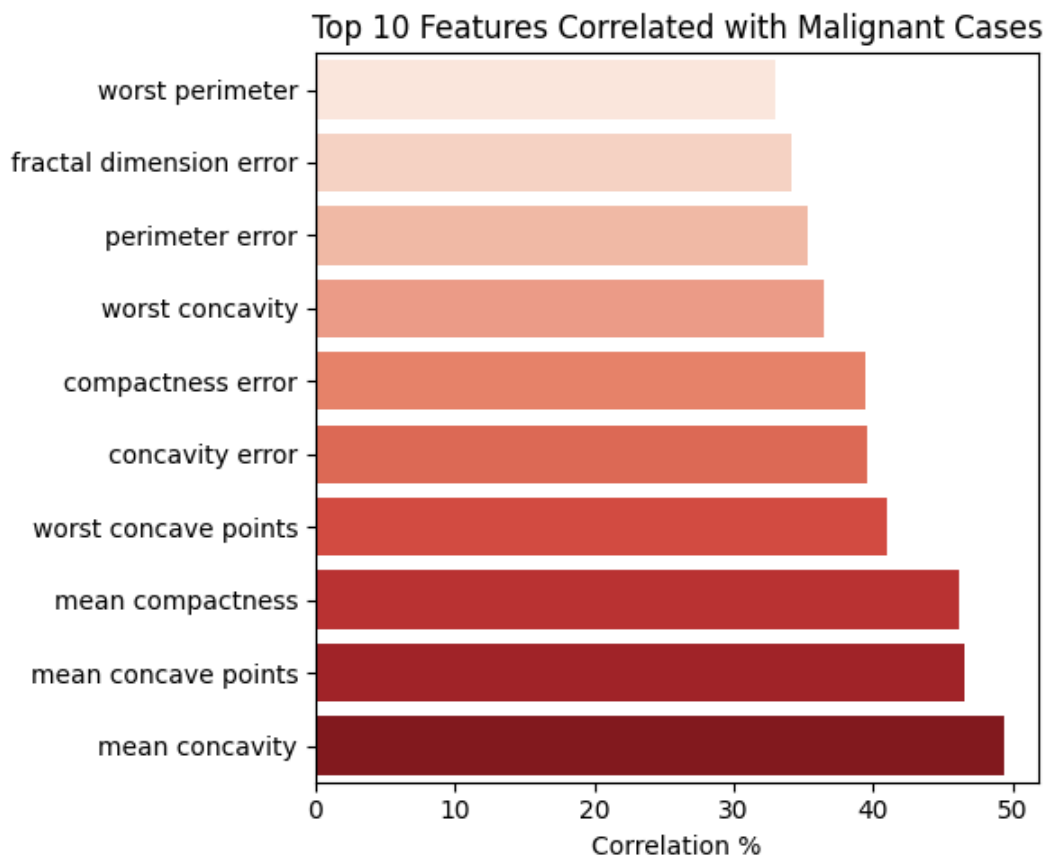
# Calculating the correlation matrix for malignant cases
correlation_matrix_malignant = df_malignant.corr()

# Sorting the correlations with the mean correlation of all features
correlations_malignant_sorted = correlation_matrix_malignant.mean().sort_values()

# Top 10 features based on correlation with 0: Malignant cases
top_10_features = correlations_malignant_sorted.tail(10)

# Converting correlations to percentages
top_10_features_percentage = top_10_features * 100

# Plotting bar chart with gradient colors based on percentage values
plt.figure(figsize=(6,5))
sns.barplot(y=top_10_features_percentage.index, x=top_10_features_percentage.values, palette="
plt.title("Top 10 Features Correlated with Malignant Cases")
plt.xlabel("Correlation %")
plt.tight_layout()
plt.show()
```



## Summary and Conclusions

### 1. Summary of Key Findings

**Positive Correlation:** Features such as **mean concavity**, **mean perimeter**, **mean area**, and **mean radius** show high positive correlations, indicating that as these features increase, the likelihood of a tumor being malignant also increases.

**Negative Correlation:** Conversely, features like **mean smoothness**, **worst fractal dimension**, and **mean symmetry** exhibit negative correlations with the target, indicating that higher smoothness and symmetry values are associated with benign tumors.

**Overall:** The **geometric properties and size-related features** of the tumor are more strongly associated with malignancy, highlighting their importance in distinguishing malignant tumors from benign ones.

---

## 2. What are the key takeaways from the exploratory data analysis?

- The analysis of the cancer dataset highlights key features that are crucial for distinguishing between malignant and benign tumors (**Area, Radius, Perimeter and Concavity**).
- 

## 3. Recommendations For Further Analysis:

- By focusing on the significant features, develop effective predictive models and diagnostic tools to aid in early detection and improve patient outcomes.
- Exclude uncorrelated features from the models to reduce complexity and improve performance.
- Further analysis and validation are necessary to ensure the clinical relevance of these findings.

Thank you...

---

Duygu Jones | Data Scientist | May 2024

Connect with me: [duygujones.com](https://duygujones.com) | [Linkedin](#) | [GitHub](#) | [Kaggle](#) | [Medium](#) | [Tableau](#)