Exercise 1.2

1. Model Construction:

In [ ]:
```
import pandas as pd

df = pd.read_csv('Files_For_A2/cancer_data.csv')
df.head()
```

Out[ ]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	sr
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

5 rows × 32 columns

In [ ]:
```
df.drop('id', axis=1, inplace=True)
```

In [ ]:
```
# convert the categorical data to numerical data
from sklearn.preprocessing import LabelEncoder

# initialize LabelEncoder
labelencoder = LabelEncoder()

# convert the categorical data to numerical data and display the first 5 row
df['diagnosis'] = labelencoder.fit_transform(df['diagnosis'])
df
```

Out[ ]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothnes
0	1	17.99	10.38	122.80	1001.0	
1	1	20.57	17.77	132.90	1326.0	
2	1	19.69	21.25	130.00	1203.0	
3	1	11.42	20.38	77.58	386.1	
4	1	20.29	14.34	135.10	1297.0	
...	...	...	...	...	...	
564	1	21.56	22.39	142.00	1479.0	
565	1	20.13	28.25	131.20	1261.0	
566	1	16.60	28.08	108.30	858.1	
567	1	20.60	29.33	140.10	1265.0	
568	0	7.76	24.54	47.92	181.0	

569 rows × 31 columns

In [ ]:
```
from sklearn.tree import DecisionTreeClassifier

# split the data into features and target
X = df.drop('diagnosis', axis=1)
y = df['diagnosis']

# initialize the DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X, y)
```

Out[ ]:

▾ DecisionTreeClassifier

DecisionTreeClassifier()

In [ ]:
```
feature_importances = pd.DataFrame(classifier.feature_importances_, index=cl
feature_importances['cumsum'] = feature_importances['importance'].cumsum()
feature_importances.head(10)
```

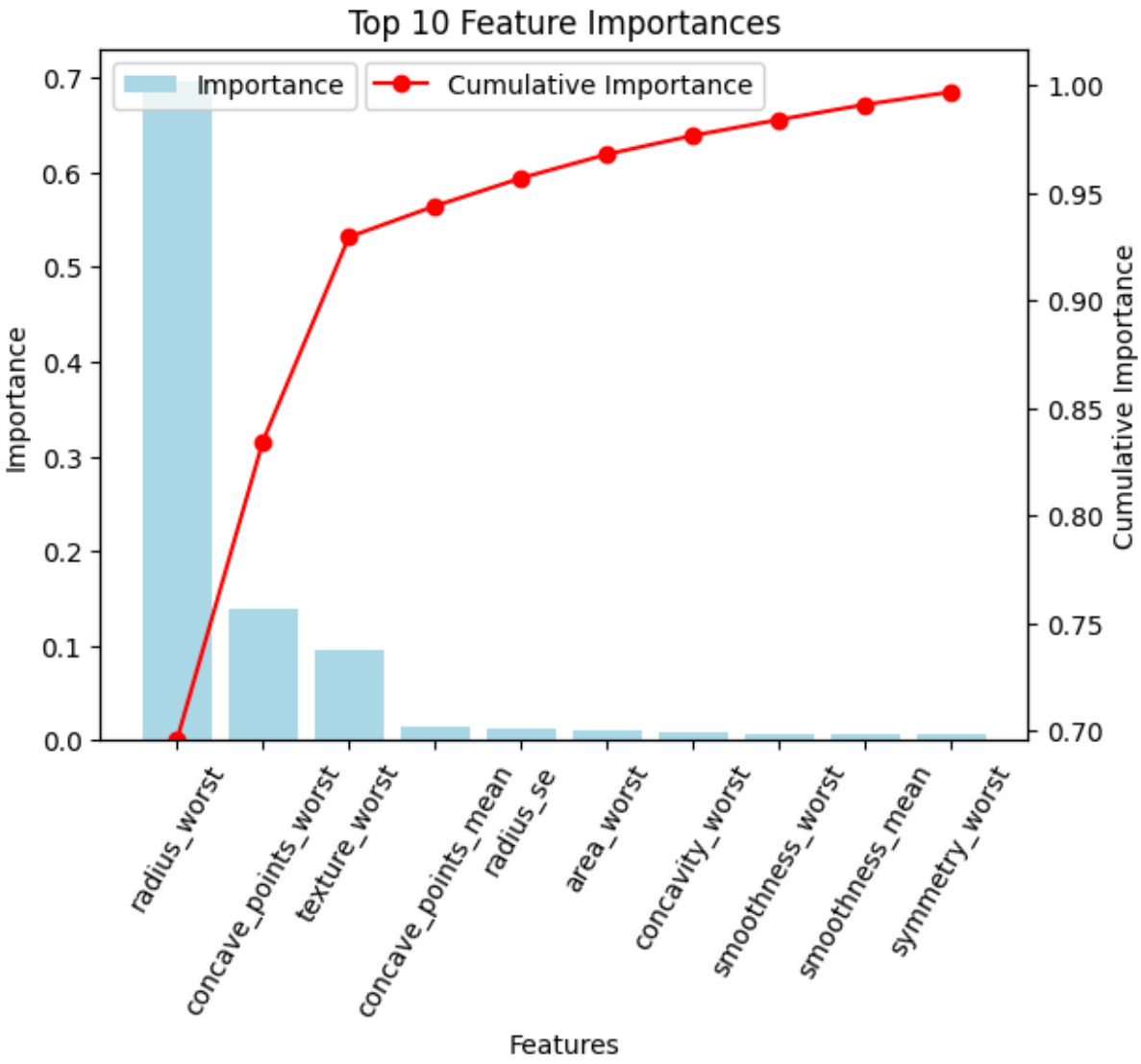
Out[ ]:

	importance	cumsum
radius_worst	0.695594	0.695594
concave_points_worst	0.138938	0.834532
texture_worst	0.095005	0.929537
concave_points_mean	0.014410	0.943947
radius_se	0.012955	0.956901
area_worst	0.011086	0.967987
concavity_worst	0.008727	0.976715
smoothness_worst	0.007388	0.984103
smoothness_mean	0.007017	0.991120
symmetry_worst	0.005831	0.996951

In [ ]:
```
import matplotlib.pyplot as plt

# Create a figure and axis objects
fig, ax1 = plt.subplots()

# Plot the bar chart on the first y-axis
ax1.bar(feature_importances.index[:10], feature_importances['importance'][:1

# Set the x-axis label
ax1.set_xlabel('Features')

# Set the first y-axis label
ax1.set_ylabel('Importance')

# Set the title
ax1.set_title('Top 10 Feature Importances')

ax1.set_xticklabels(feature_importances.index[:10], rotation=60)

# Create a second y-axis
ax2 = ax1.twinx()

# Plot the line graph on the second y-axis
ax2.plot(feature_importances.index[:10], feature_importances['cumsum'][:10],

# Set the second y-axis label
ax2.set_ylabel('Cumulative Importance')

# Set the limits of the second y-axis based on the minimum and maximum value
ax2.set_ylim(feature_importances['cumsum'][:10].min(), feature_importances['

# Add a legend
ax1.legend(loc='upper left')
ax2.legend(loc='upper center')
```

```
# Show the plot
plt.show()
```

```
/var/folders/2v/mcgfxq4d2_n2639c1xhyd92w0000gn/T/ipykernel_35793/1564416163.
py:18: UserWarning: set_ticklabels() should only be used with a fixed number
of ticks, i.e. after set_ticks() or using a FixedLocator.
  ax1.set_xticklabels(feature_importances.index[:10], rotation=60)
```



4. Analysis

From the Top 10 Feature Importances graph it is clear that the radius_worst feature has the most significant influence of the models predictions. After that, the importance scores decrease significantly, with concave_points_worst and texture_worst being the next 2 most important features. The cumulative importance curve shows us that adding more features beyond the top 1 - 3 will provide diminishing returns in terms of the models performance.

To develop more accurate models using these insights, we can focus on the most important features and ensure they are accurate and well-preprocessed because errors

in these feautures will have a greater impact on model performance. Also, since there is a steep dropoff in feature importance we can use dimensionality reduction techniques such as PCA without losing significant reliability.

Overall, it is clear that the insights from the Feature Importance graph suggest that there is a significant drop in the influence of features beyond 1-3. This would have a significant impact on the development of more accurate predictive models because we can allocate resources and focus on optimizing the most predictive features.