Problem definition:

Investigate whether the provided features can help predict the diagnosis (Malignant or Benign) of breast cancer.

# CRISP-DM methodology:

## a. Business understanding:

In this project, we aim to create a data visualization/dashboard that addresses a specific question related to breast cancer diagnosis. The clearly defined topic we will focus on is:

Can the geometric properties of cell nuclei in breast cancer patients accurately predict the diagnosis (Malignant or Benign) and help facilitate decision-making for medical professionals?

To address this question, we will develop a data visualization/dashboard that illustrates the relationships between the geometric properties of cell nuclei and breast cancer diagnosis. This visualization will provide medical professionals with insights that can inform their decision-making process when diagnosing and treating breast cancer patients.

**Key stakeholders for this project include:**

Doctors and oncologists: These medical professionals can use the visualization to better understand the relationship between cell nuclei properties and breast cancer diagnosis, which may inform their decisions regarding patient treatment and care.

Patients: Patients can benefit from more accurate and informed diagnoses, leading to better treatment outcomes.

Medical researchers: Researchers can utilize the visualization to identify patterns and trends in the data, potentially guiding further research and development of new diagnostic methods or treatments.

By addressing this specific question, our data visualization/dashboard aims to support the decision-making process for medical professionals and contribute to the overall goal of improving breast cancer diagnosis and treatment outcomes.

## b. Data understanding:

To gain insights into the provided features and understand the relationships between features and the target variable (diagnosis), we will perform the following steps:

**Data exploration:**

**Tried to do a full EDA:**



But the html file is to large to view in the browser or notebook, rendering it less useful.  
  
**Then went to a manual approach:**

Examine the dataset to get an overview of the features and their data types.

Calculate descriptive statistics for the features, such as mean, median, and standard deviation, to understand the central tendency and dispersion of the data.

Identify any missing values or inconsistencies in the data that may need to be addressed during the data preparation phase.

**Visualization:**

Create histograms or box plots for each feature to visualize the distribution of values and identify any potential outliers or patterns.

Generate scatter plots or pair plots to visualize relationships between pairs of features, which can help identify potential correlations or interactions.

Create a correlation matrix or heatmap to quantify the relationships between the features and the target variable (diagnosis).

**Feature-target relationship analysis:**

Examine the distribution of the target variable (diagnosis) to understand the balance between the two classes (Malignant and Benign).

Analyze the relationships between individual features and the target variable using visualizations such as bar plots or violin plots, which can help identify features that have strong associations with the diagnosis.

By executing these steps in the data understanding phase, we will gain a comprehensive understanding of the dataset's features and their relationships with the breast cancer diagnosis. This knowledge will inform our decisions during the data preparation and modeling phases, ensuring the development of an accurate and effective data visualization/dashboard.

## Taking into account data\_description.txt, data\_head.txt, data\_missing.txt we analysed the dataset and its features.

**Target variable distribution**: The diagnosis\_numeric statistics indicate that about 37.26% of the samples have a diagnosis of M (malignant), while the remaining 62.74% have a diagnosis of B (benign). This suggests that the dataset is moderately imbalanced, which might lead to biased model predictions. It is essential to consider techniques like resampling, stratified sampling, or using appropriate evaluation metrics to handle this imbalance.

**Feature distributions**: A quick glance at the first five rows of data provides an initial understanding of the feature distributions. We notice that features like radius\_mean, texture\_mean, and area\_mean have different ranges and scales, reaffirming the need for feature scaling during preprocessing. Additionally, some features appear to have a skewed distribution, like concavity\_mean and concave points\_mean. These may require data transformations to address the skewness.

**Feature importance**: It is important to analyze the relationships between the features and the target variable (diagnosis) to identify the most relevant and influential features. Techniques like correlation analysis, mutual information, or feature importance from decision-tree-based models can help identify the most important features for the classification task.

**Multicollinearity**: Some features might be highly correlated, leading to multicollinearity, which can adversely affect certain machine learning algorithms. For example, radius\_mean, perimeter\_mean, and area\_mean are expected to have a high correlation as they all relate to the size of the cell nucleus. It is crucial to analyze the correlation between features and consider dimensionality reduction techniques like PCA (Principal Component Analysis) or feature selection methods to handle multicollinearity.

**Outliers**: Outliers may affect the performance of certain machine learning algorithms. Investigating the dataset for potential outliers and deciding on a strategy to handle them (e.g., removing, capping, or applying robust scaling) is essential for building accurate and reliable models.

**In conclusion**: To build an effective predictive model, it is essential to address the dataset's challenges, such as class imbalance, feature scaling, skewed distributions, multicollinearity, and outliers. By applying appropriate preprocessing and feature engineering techniques, it is possible to enhance the dataset's quality and improve the performance of the machine learning models.

## c. Data preparation:

Pre-process the data: handle missing values, outliers, and data imbalances.

# Problems were encountered, scale\_features was losing the target feature,

# first separate the features and the target variable, apply the scaling only to the features, and then combine the scaled features with the target variable. This way, the target variable remains unchanged.

# first Separate the features and target variable before applying PCA.

Concatenate the transformed features with the target variable after applying PCA

However, it's important to note that the interpretability of the models might be limited, as the principal components are not directly related to the original features.

Missing values is in the handle\_outliers function. When outliers are removed, the indices of the DataFrame are not reset, which can cause issues when merging data.





Feature engineering: create new features or transform existing ones if needed.

Split the dataset into training and testing sets.

## d. Modeling (only for Data Science (Eng) 874 students):

Select appropriate machine learning algorithms for classification.

Train and optimize the models using the training set.

Evaluate model performance using the testing set.

To select appropriate machine learning algorithms for classification, we can choose a few popular classifiers and evaluate their performance. In this example, we will use Logistic Regression, Support Vector Machine (SVM), Random Forest, and Gradient Boosting. We will train and optimize the models using the training set and evaluate their performance using the testing set.

First, let's split the dataset into training and testing sets.

Next, let's train and optimize the models:

Logistic Regression

Support Vector Machine

Random Forest

Gradient Boosting

Tune hyperparameters using GridSearchCV

## e. Evaluation:

Now let's create a function to evaluate each model: evaluate\_model

## Here are the performance metrics for each model:

## Logistic Regression:

## Accuracy: 0.9492

## Precision: 0.9815

## Recall: 0.9138

## F1 Score: 0.9464

## Support Vector Machine:

## Accuracy: 0.9407

## Precision: 0.9636

## Recall: 0.9138

## F1 Score: 0.9381

## Random Forest:

## Accuracy: 0.9322

## Precision: 0.9464

## Recall: 0.9138

## F1 Score: 0.9298

## Gradient Boosting:

## Accuracy: 0.9407

## Precision: 0.9474

## Recall: 0.9310

## F1 Score: 0.9391

## Based on these results, the Logistic Regression model performs the best with the highest accuracy, precision, and F1 score. Although the recall is slightly lower than that of the Gradient Boosting model, its overall performance is still better.

## f. Deployment:

Create an interactive dashboard to visualize the results and predictions.

Share the dashboard and report with relevant stakeholders.

Visualization/dashboard:

Create visualizations that highlight relationships between features and diagnosis.

Incorporate the predictive model (for Data Science (Eng) 874 students) into the dashboard to display future predictions.

Ensure the dashboard is user-friendly and follows Tufte's Visualization Aesthetic principles.

Report:

Document each step of the CRISP-DM process in the report.

For Data Science (Eng) 874 students, include a section explaining the chosen prediction algorithm and its performance.

Submit the report in PDF format along with the data visualization/dashboard.