**Breast Cancer prediction**

### **Data Preprocessing:**

* 1. Import Libraries:
  + Pandas is used for data manipulation and analysis.
  + train\_test\_split from sklearn.model\_selection is used for splitting the dataset.
  + StandardScaler is used to standardize features.
  + PolynomialFeatures is used to generate polynomial features.
  + SVC from sklearn.svm is used for Support Vector Classification.
  + Various metrics from sklearn.metrics are imported for model evaluation.
* 2. Upload Dataset:
  + The dataset is uploaded manually to Colab by using the Colab interface.
* 3. Load Dataset:
  + The dataset is loaded into a Pandas DataFrame using pd.read\_csv.
  + The 'id' column is dropped as it seems unnecessary.
* 4. Handling Missing Values:
  + Missing values are checked and any rows with missing values are dropped.
* 5. Mapping Target Variable:
  + The 'diagnosis' column (categorical) is mapped to numerical values: 'M' to 1 and 'B' to 0.
* 6. Check for Outliers:
  + Outliers are identified using a z-score method. If the z-score is greater than 3 or less than -3, the data point is considered an outlier.
  + Outliers are printed for each feature.
* 7. Handle Outliers:
  + Rows containing outliers are removed from the dataset.
* 8. Feature Engineering:
  + A new feature is created by multiplying 'mean\_radius' and 'mean\_texture' columns.
* 9. Introduce Polynomial Features:
  + Polynomial features of degree 2 are introduced using PolynomialFeatures from sklearn.preprocessing.

### **Model Implementation:**

* 1. Split the Dataset:
  + The dataset is split into training and testing sets using train\_test\_split.
* 2. Standardize Features:
  + Standardization is applied to the features using StandardScaler.
* 3. Create and Train SVM Model:
  + An SVM model with a linear kernel is created and trained on the training data.
* 4. Make Predictions:
  + Predictions are made on the test set.
* 5. Evaluate the Model:
  + Accuracy, confusion matrix, and classification report are printed to evaluate the model's performance.

**Model Performance Metrics:**

### Accuracy:

### Accuracy is calculated as the ratio of correctly predicted observations to the total observations.

### Accuracy = (True Positives+True Negatives)/Total Observations​

### In this case, the accuracy is ≈ 94.74%

1. Confusion Matrix:

* The confusion matrix provides a breakdown of correct and incorrect predictions
* Here, 67 instances of class 0 are correctly predicted, 4 instances of class 0 are incorrectly predicted as class 1, 2 instances of class 1 are incorrectly predicted as class 0, and 41 instances of class 1 are correctly predicted.

1. Classification Report:

* Precision, recall, and F1-score are metrics for both classes (0 and 1).
* Macro Avg provides the unweighted average across classes.
* Weighted Avg provides the weighted average based on the number of instances in each class.

### **Challenges:**

1. Handling Missing Values and Outliers:
   * Missing values and outliers are removed from the dataset, which may lead to a loss of information. Alternative strategies like imputation or robust models could be explored.
2. Model Selection:
   * The choice of a linear SVM might be limiting. Experimenting with different kernels and tuning hyperparameters could improve the model.
3. Feature Engineering:
   * Feature engineering is performed by creating a new feature and introducing polynomial features. The impact of these features on the model's performance should be analyzed.