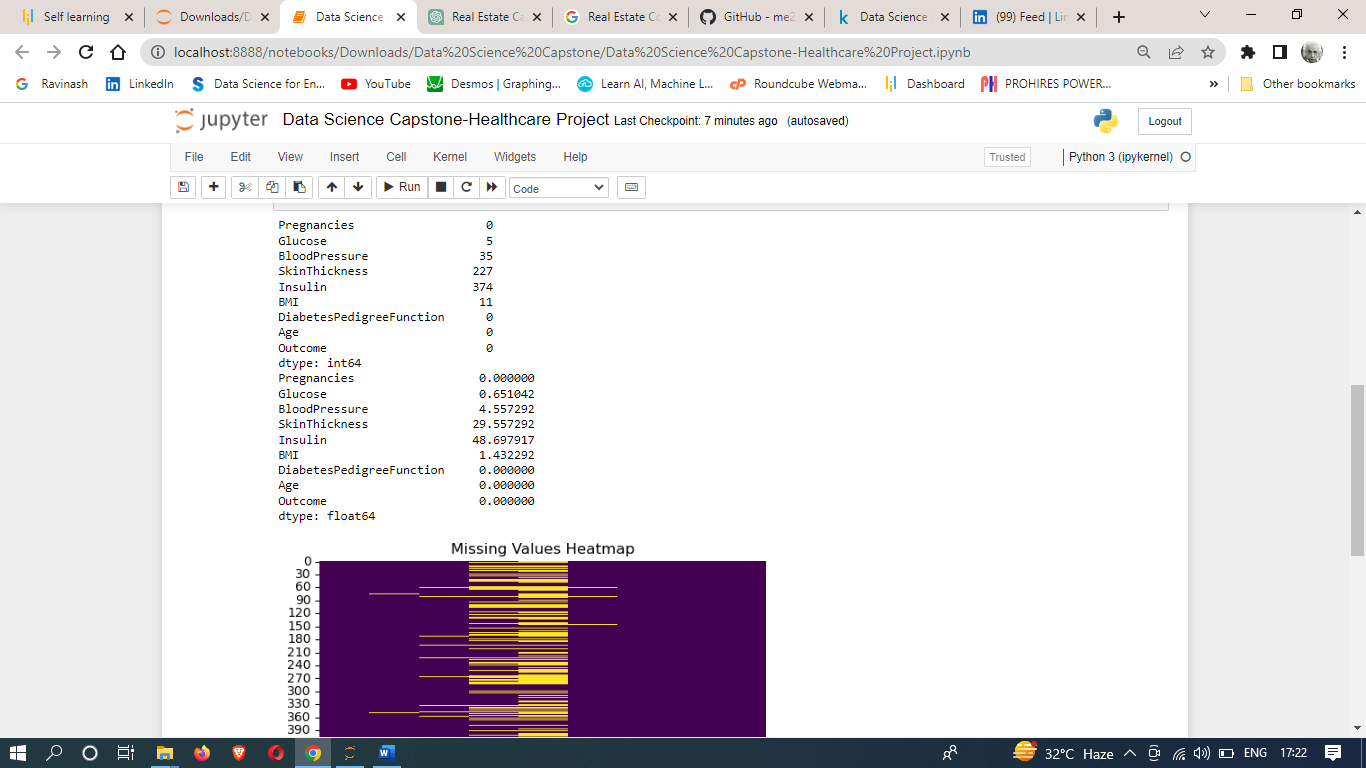
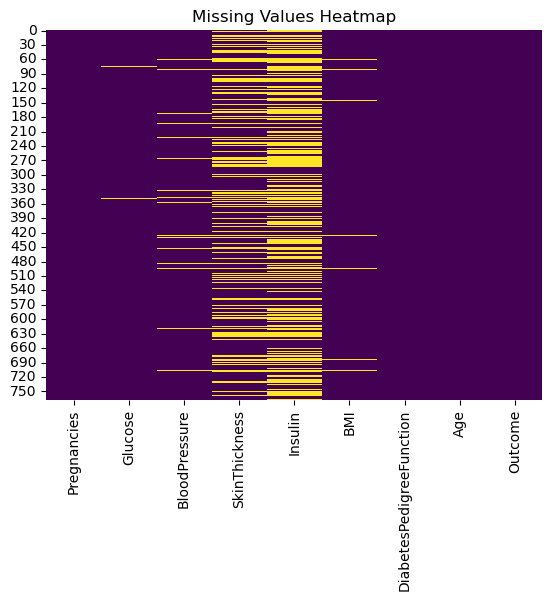
**Data Exploration:**

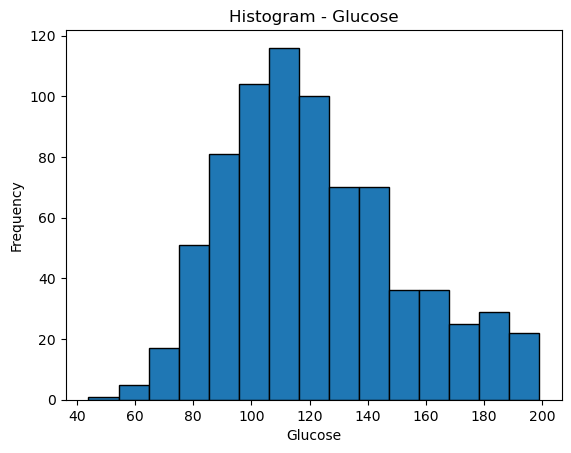
1. **Perform descriptive analysis. Understand the variables and their corresponding values. On the columns below, a value of zero does not make sense and thus indicates missing value:**

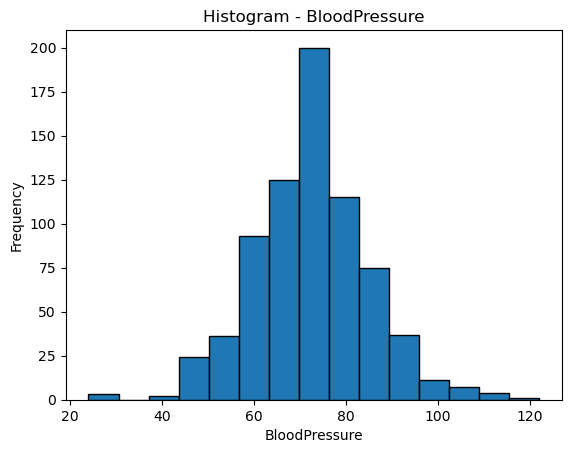
* **Glucose**
* **BloodPressure**
* **SkinThickness**
* **Insulin**
* **BMI**

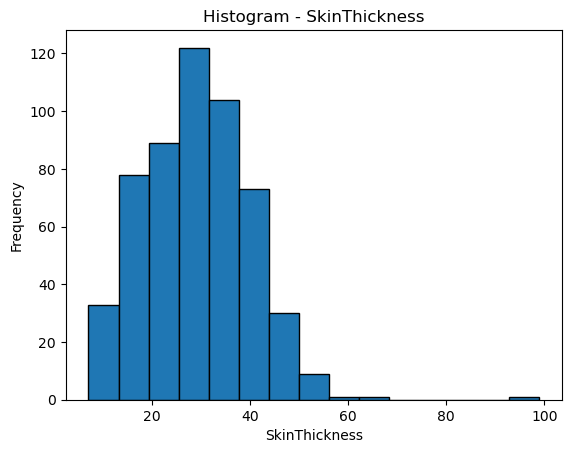


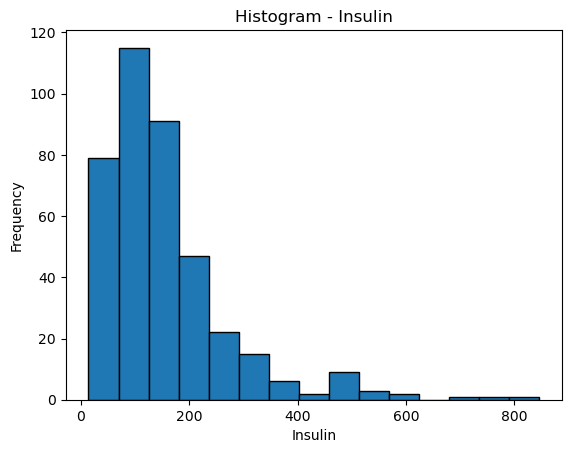


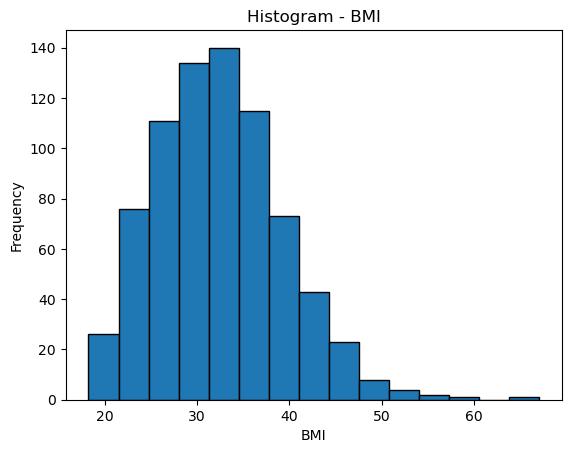
1. **Visually explore these variables using histograms. Treat the missing values accordingly.**



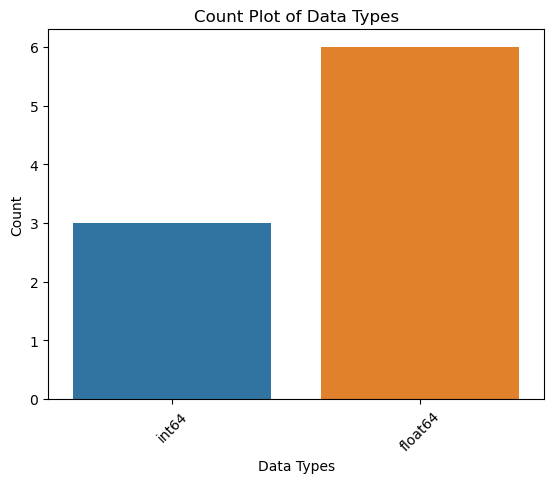




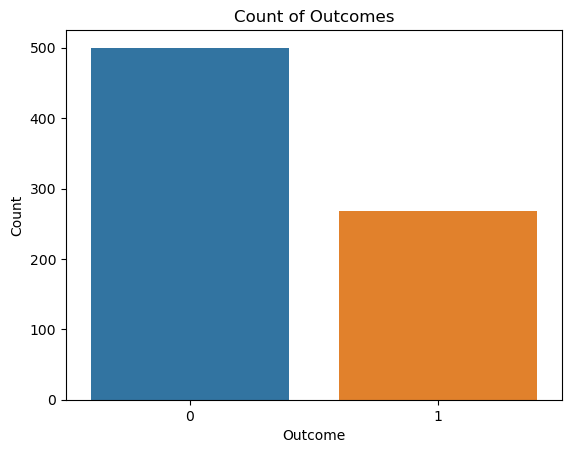




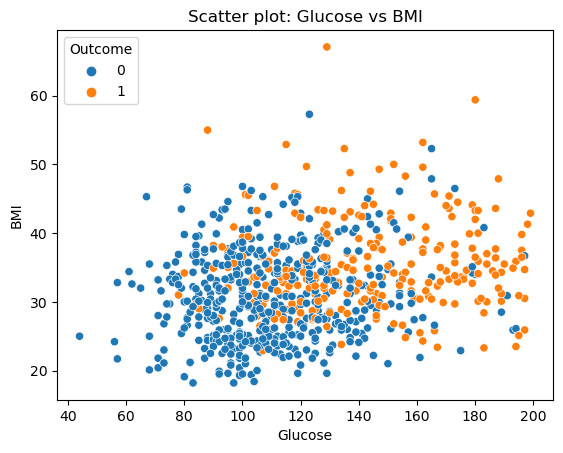
1. **There are integer and float data type variables in this dataset. Create a count (frequency) plot describing the data types and the count of variables.**

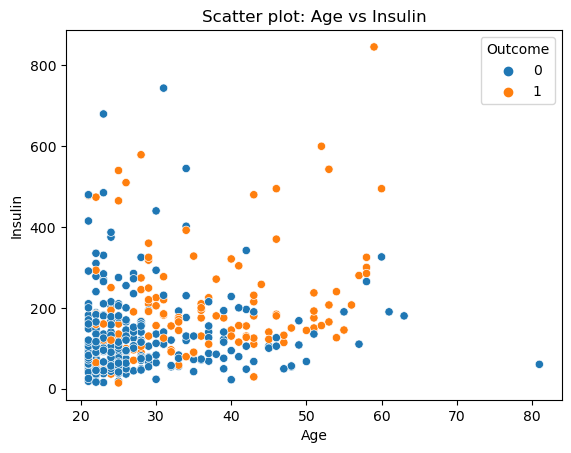
****

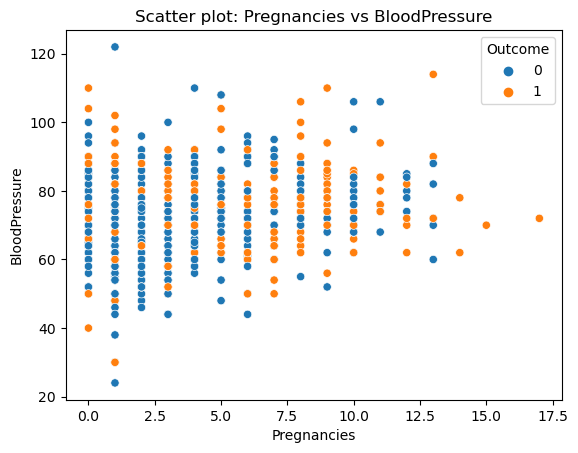
1. **Check the balance of the data by plotting the count of outcomes by their value. Describe your findings and plan future course of action.**

****

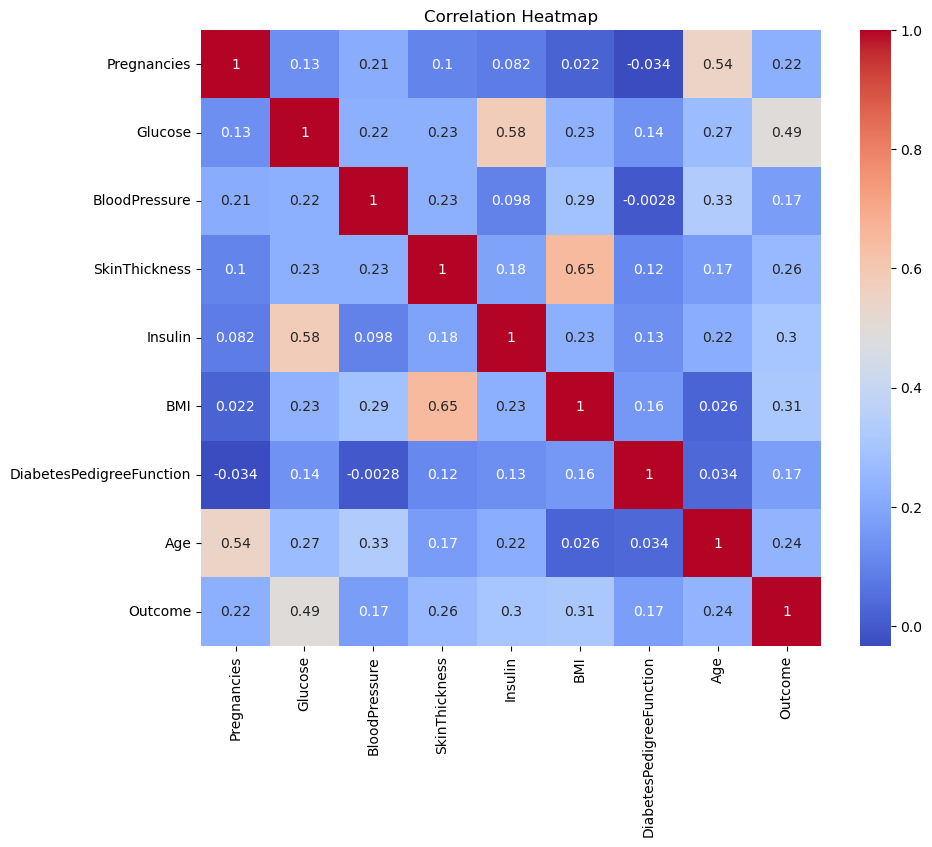
1. **Create scatter charts between the pair of variables to understand the relationships. Describe your findings.**

****

****

****

1. **Perform correlation analysis. Visually explore it using a heat map**.



**Week 2**

**Data Modeling:**

1. **Devise strategies for model building. It is important to decide the right validation framework. Express your thought process.**

**Ans. When building a predictive model, it is crucial to follow a systematic approach and choose an appropriate validation framework. Here are some strategies and considerations for model building:**

**1.** Splitting the Data: Divide the dataset into training and testing sets. The training set will be used to train the model, while the testing set will be used to evaluate its performance on unseen data. The usual practice is to use a 70-30 or 80-20 split, but this can vary depending on the size of the dataset.

2. Handling Class Imbalance: Since the dataset has a class variable (Outcome) with imbalanced classes (more 0s than 1s), it's important to address this issue during model building. Techniques like oversampling the minority class or undersampling the majority class can be employed to balance the dataset. Another approach is to use evaluation metrics that are less sensitive to class imbalance, such as area under the ROC curve (AUC-ROC) or precision-recall curve.

3. Selecting Evaluation Metrics: Choose appropriate evaluation metrics that align with the problem and reflect the model's performance. For binary classification problems like predicting diabetes (0 or 1), metrics such as accuracy, precision, recall, F1-score, and AUC-ROC can be used. However, due to the class imbalance, accuracy alone may not be sufficient. Consider metrics that give a more comprehensive view of the model's performance, particularly its ability to correctly identify positive cases (diabetes).

4. Feature Selection: Evaluate the importance and relevance of each feature in the dataset. Remove any redundant or irrelevant features that may negatively impact the model's performance or introduce noise. Techniques like correlation analysis, feature importance from tree-based models, or regularization methods (e.g., L1 regularization) can aid in feature selection.

5. Model Selection: Consider various algorithms suitable for binary classification, such as logistic regression, decision trees, random forests, support vector machines (SVM), or gradient boosting algorithms (e.g., XGBoost, LightGBM). Compare their performance using appropriate validation techniques (e.g., cross-validation) and select the algorithm that provides the best results based on the chosen evaluation metrics.

6. Hyperparameter Tuning: Fine-tune the selected model by optimizing its hyperparameters. Hyperparameters control the behavior of the model and can greatly impact its performance. Techniques like grid search, random search, or Bayesian optimization can be used to find the optimal combination of hyperparameters that maximize the model's performance.

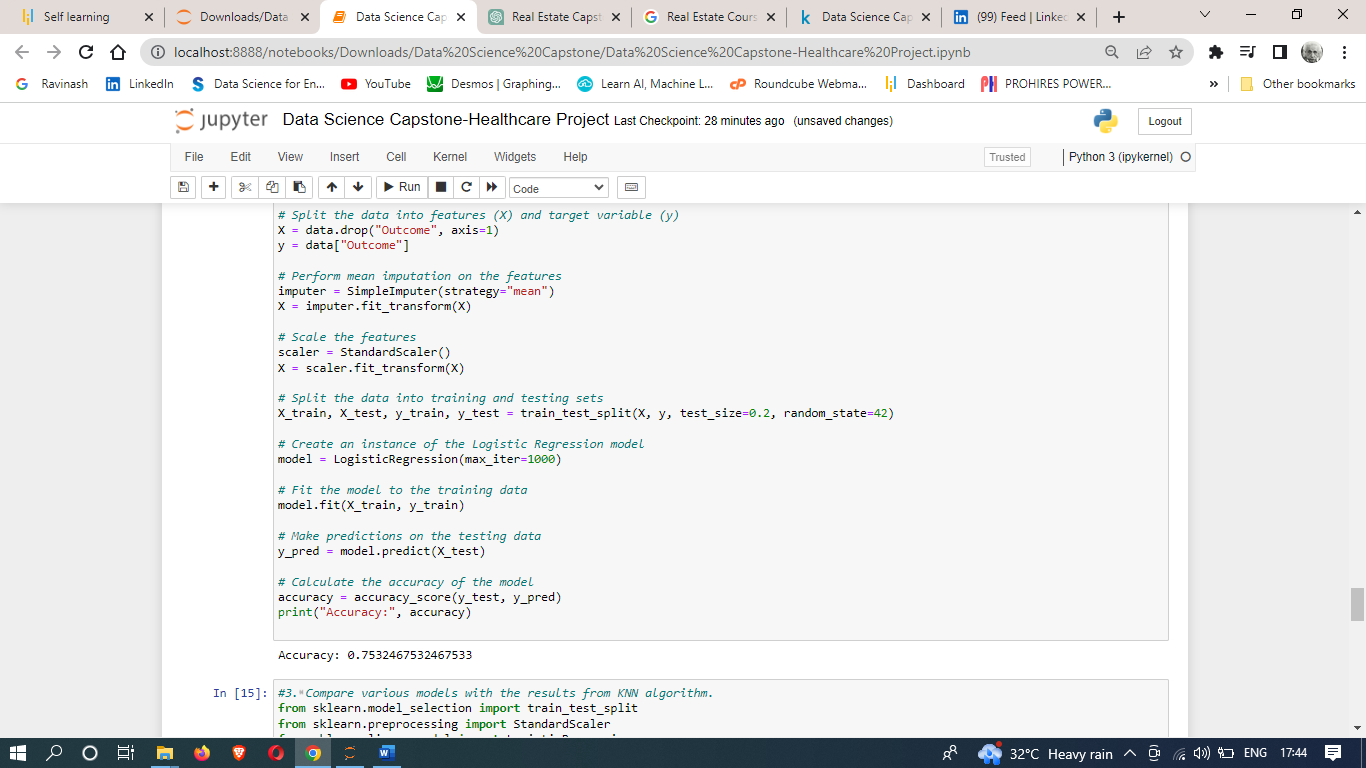
7. Model Evaluation and Validation: Once the final model is trained and tuned, evaluate its performance on the testing set. Use appropriate validation techniques to ensure that the model's performance is consistent and not overfit to the training data. Techniques like k-fold cross-validation can provide a more robust estimate of the model's performance.

8. Interpretability and Explainability: Consider the interpretability and explainability of the chosen model. Some models, like decision trees or logistic regression, provide more transparency and can help in understanding the factors influencing the prediction. This can be important for gaining insights, building trust, and meeting regulatory requirements.

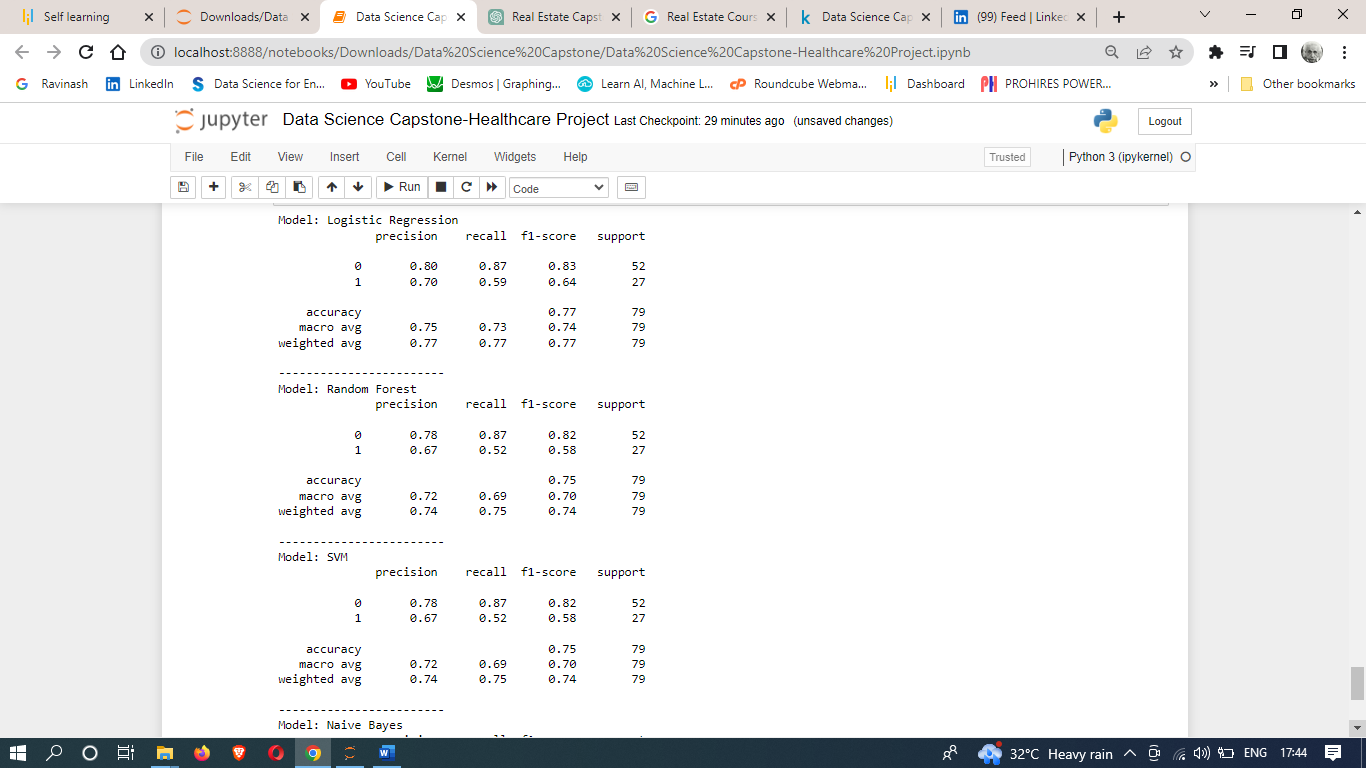
By following these strategies, we can develop a robust and reliable predictive model for diabetes classification. The chosen validation framework should ensure that the model performs well on unseen data and is capable of generalizing to new instances.

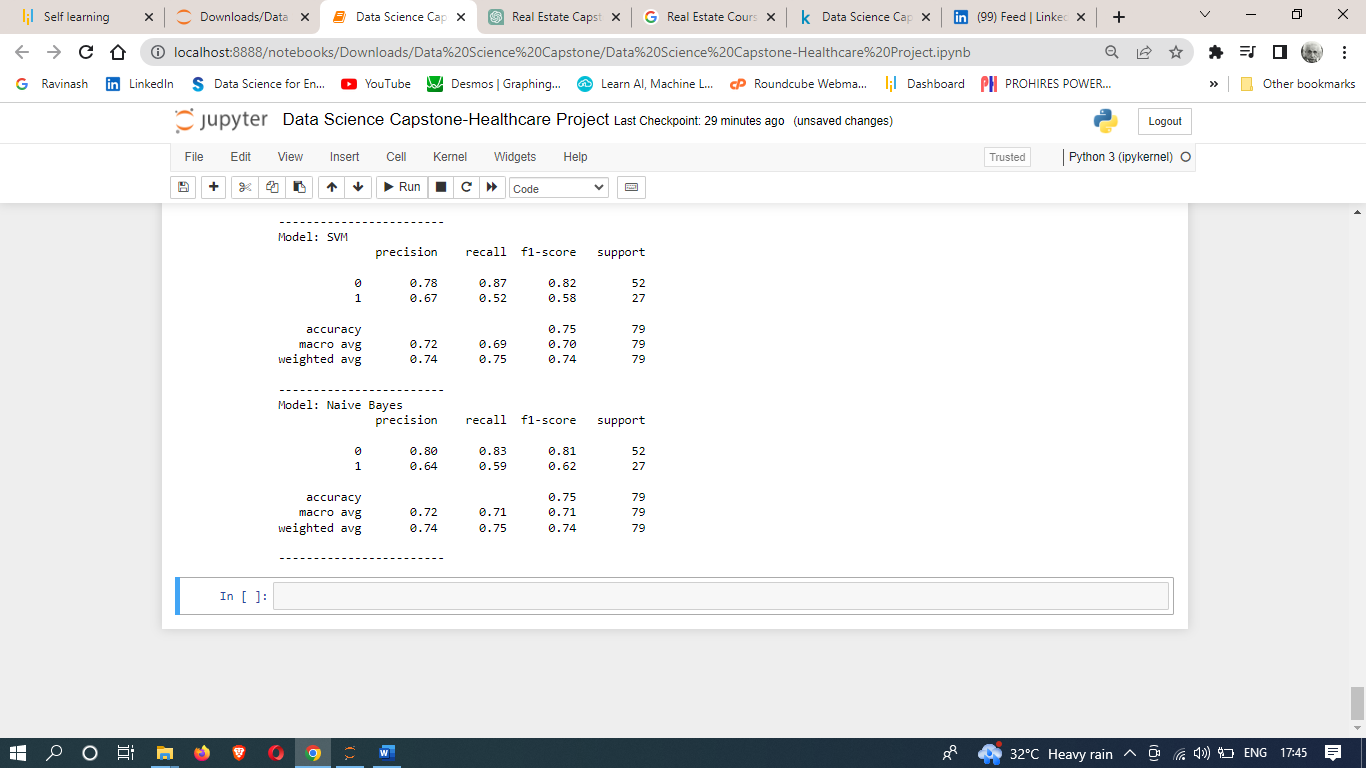
1. **Apply an appropriate classification algorithm to build a model.**

**Ans.:** Accuracy: 0.7532467532467533



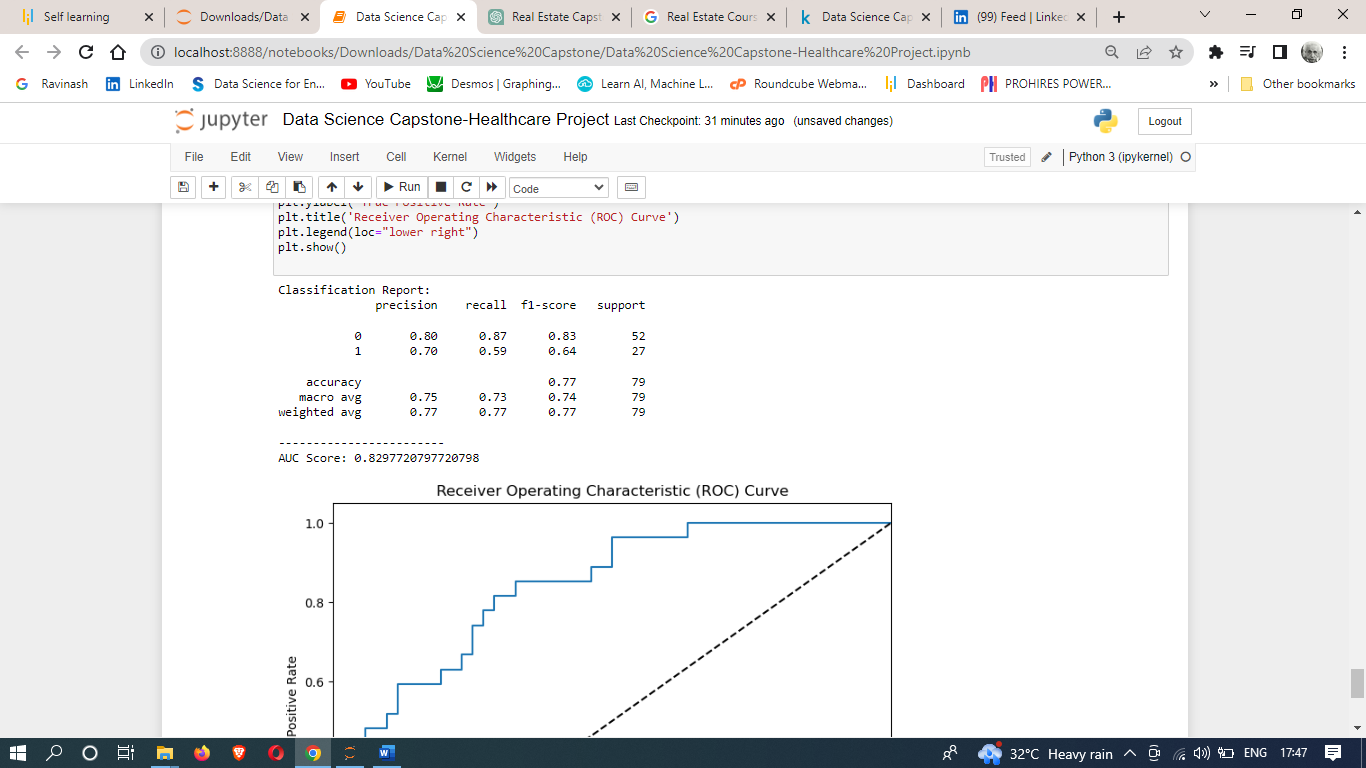
1. **Compare various models with the results from KNN algorithm.**

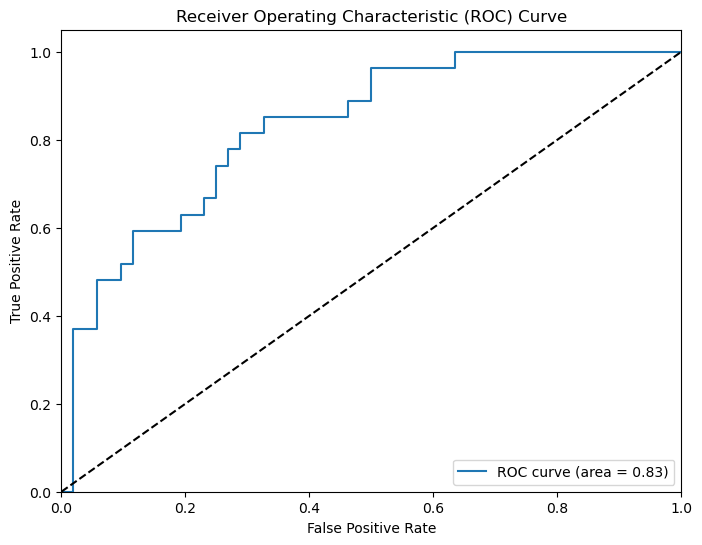




1. **Create a classification report by analyzing sensitivity, specificity, AUC (ROC curve), etc.**

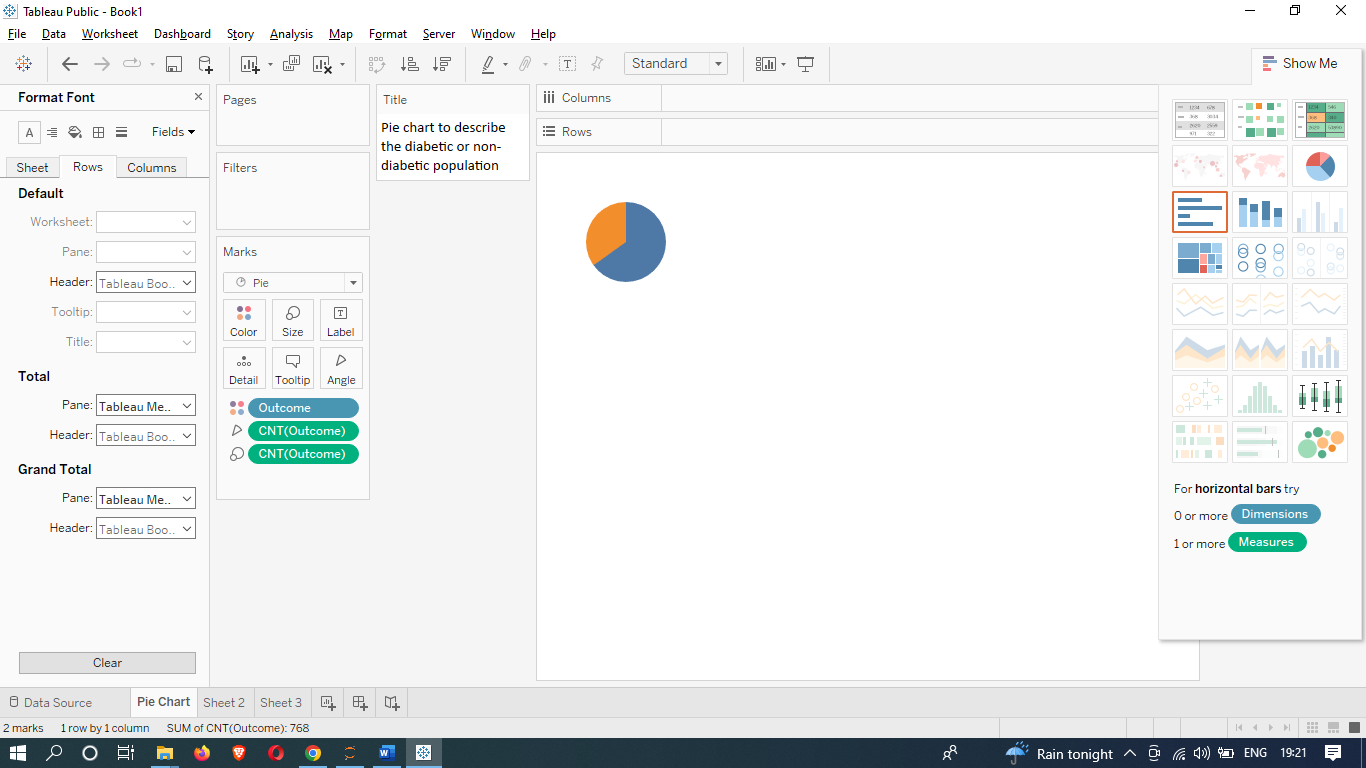
**Please be descriptive to explain what values of these parameter you have used.**



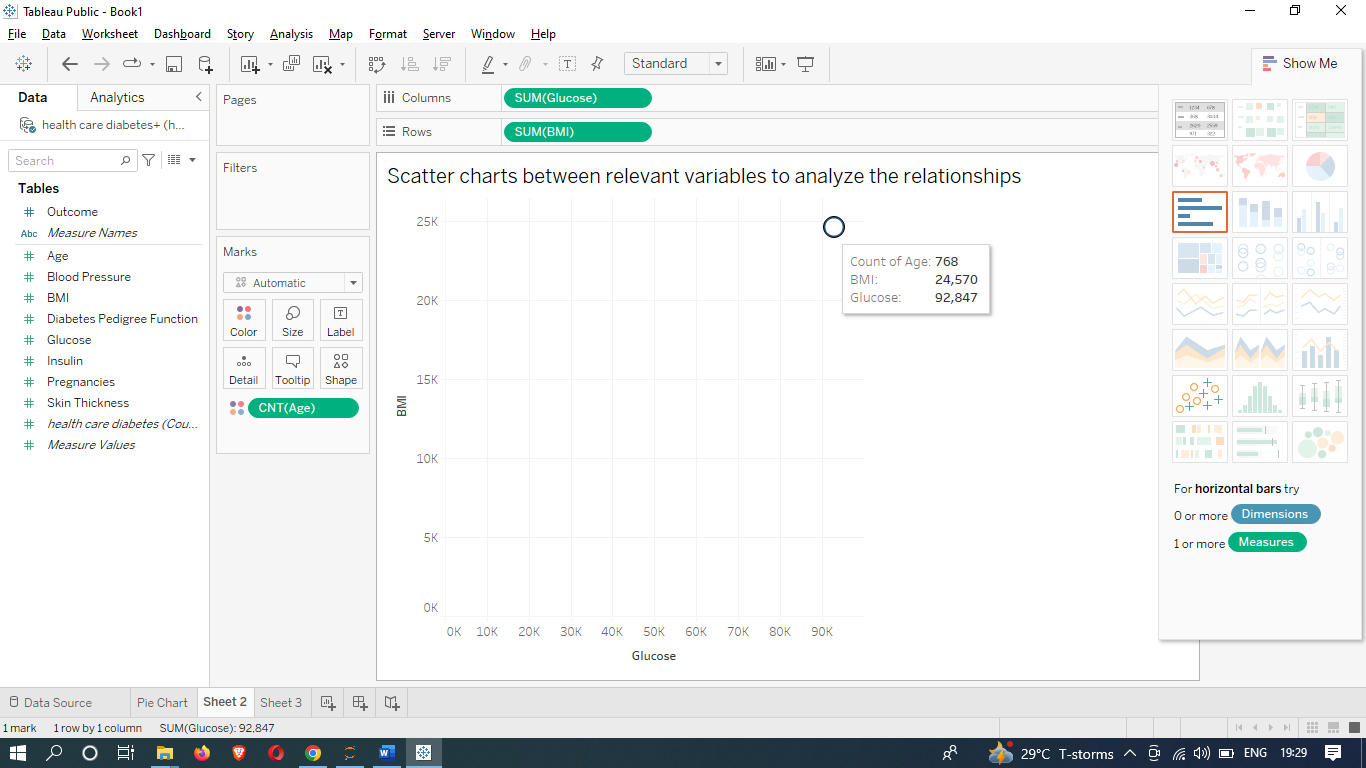
****

1. **Create a dashboard in tableau by choosing appropriate chart types and metrics useful for the business. The dashboard must entail the following:**

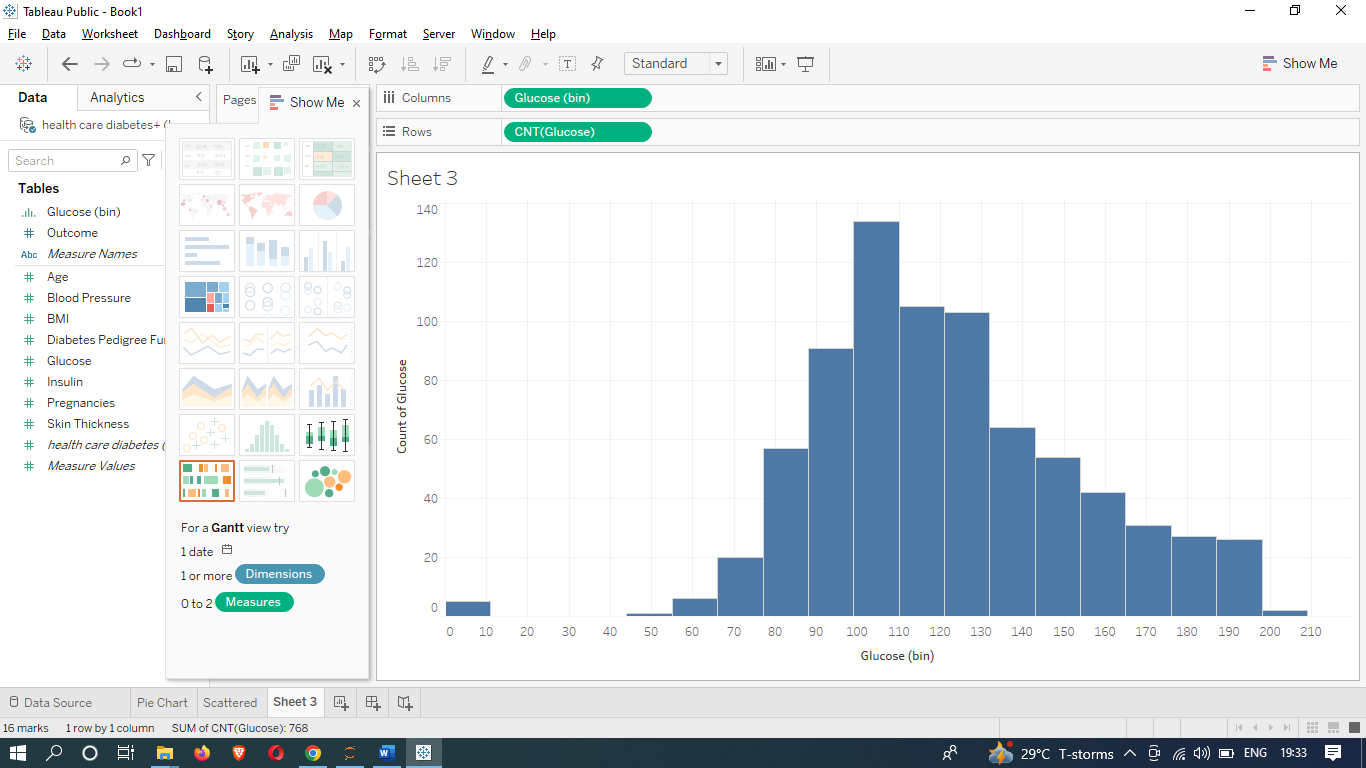
Pie chart to describe the diabetic or non-diabetic population



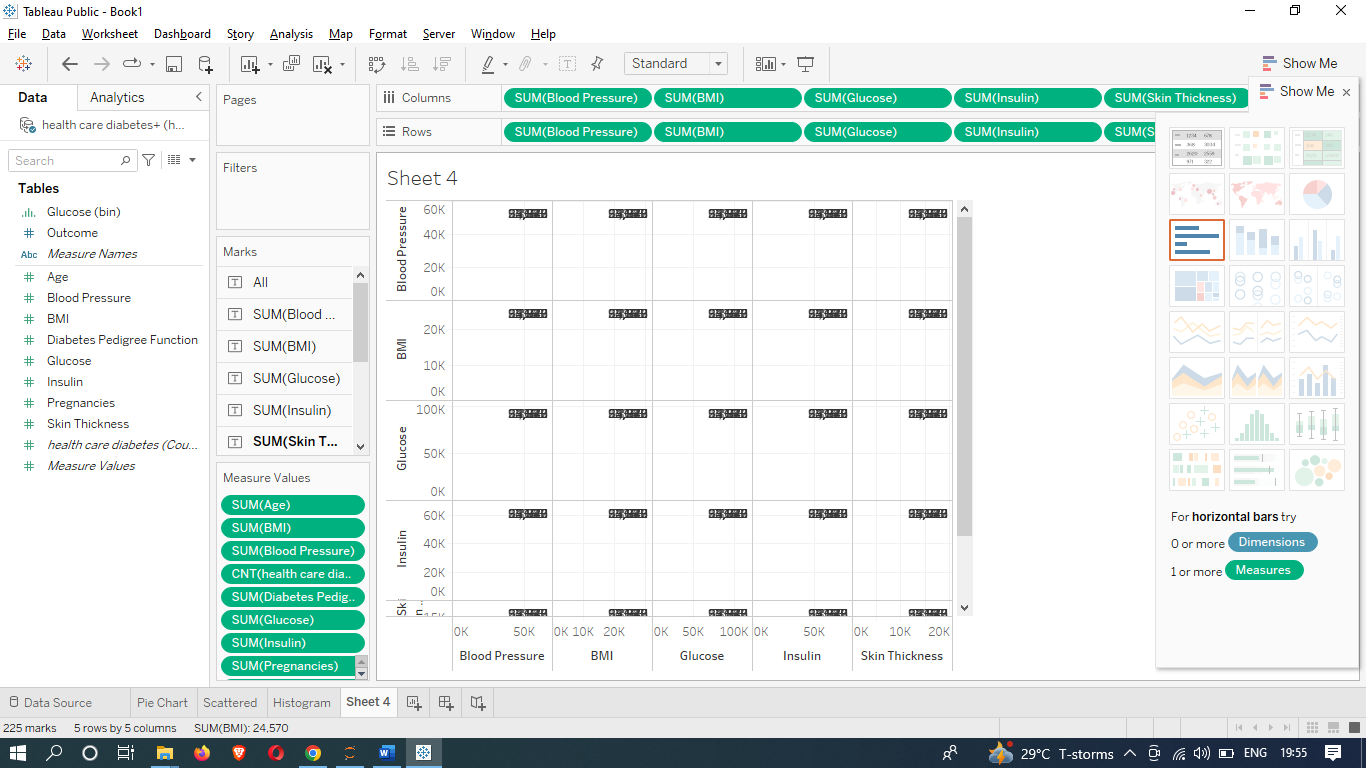
Scatter charts between relevant variables to analyze the relationships



Histogram or frequency charts to analyze the distribution of the data



Heatmap of correlation analysis among the relevant variables



Create bins of these age values: 20-25, 25-30, 30-35, etc. Analyze different variables for these age brackets using a bubble chart.

