

**Sugar consumption prediction**

**Machine learning and data mining project**

**Name: Habiba Ali Mostafa**

**Major: Applied biotechnology**

**ID: 211001893**

**Introduction:**

The aim of the project is to employ measures of sugar consumption in an attempt to predict the incidence of diabetes. The study investigates the relationship between patterns of sugar consumption and diabetes prevalence in a population according to analysis of data with variables being average daily consumption of sugar, total consumption of sugar, per capita consumption, and import/export levels. The solution includes extensive description of data cleaning, EDA, feature engineering, model building using different machine learning models, and deployment using a web interface built with Flask. Users can input the variables relating to sugar consumption into the finalized model to get the prediction for diabetes spread.

**Data Info:**

1. The df.dtypes function was used to check the data types of the dataset. What is reported is that the dataset contains 4 float columns and 1 object column. The float columns are Avg\_Daily\_Sugar\_Intake, Per\_Capita\_Sugar\_Consumption, Total\_Sugar\_Consumption, and Sugar\_Import\_Export\_Ratio. The target column, Diabetes\_Prevalence, is also a float. The object column, if present, was checked for potential encoding needs.

2. The df.info () function was used to get a summary of the data types and structure of the dataset. There are 5 variables and 100 observations in the dataset, and all of them are numeric except any possible unprocessed categorical columns (if any). The dataset appears to be well-structured for regression modeling because the dependent variable, Diabetes Prevalence, is continuous.

3. Missing values were checked using the df.isnull().sum() function. The result was that no column had any missing values, i.e., the data is clean and imputation or any other form of missing value handling is not required.

**Dataset:**

The project's dataset is about dietary consumption of sugar and how it relates to diabetes prevalence. The information is quantitative attribute that specifies numerous pieces of information about sugar consumption for populations or areas. The dataset is meant to provide a means of analyzing and predicting consumption of sugar and its impact on diabetes prevalence in a population. The dataset consists of 100 rows and 5 columns with the attributes that are as follows: • Avg\_Daily\_Sugar\_Intake: Average daily per capita intake of sugar (in grams). • Per\_Capita\_Sugar\_Consumption: Per capita consumption of sugar annually (in kilograms). • Total\_Sugar\_Consumption: Regional or countrywide total annually of sugar consumption (in metric tons). • Sugar\_Import\_Export\_Ratio: Import/export value ratio of sugar. • Diabetes\_Prevalence (target feature): Number of people with diabetes expressed as a percentage.

**Data cleaning:**

Data cleaning was the most crucial task to determine whether or not the dataset was ready for analysis and modeling for machine learning. The process below was applied:

1. Initial Inspection:

The data was then subjected to tests using functions such as df.head(), df.info(), and df.describe() in an effort to test a hypothesis on the structure of the data, to check if there were any outliers, and to check for potential errors such as missing values or invalid data types.

2. Check Missing Values

Missing values were verified in the data set using the df.isnull().sum() function. The findings were verified to be free from missing values in every column and therefore no imputation was done.

3. Data Type Verification:

All columns were assured to have appropriate data types. All columns were assured to be of float64 data type, which is sufficient for numerical analysis and regression modeling as well. There was no need for data transformation or encoding.

4. Duplicate Values:

The data were checked for duplicate rows by df.duplicated().sum(). There were no duplicate records, and thus each record is unique and sound.

5. Outlier Removal (Optional):

Data analysis using histograms and boxplots revealed that there were practically no outliers but these weren't removed as they appeared to be valid records and not an error. Outliers were not removed in order to preserve the original data distribution.

6. Column Renaming (if necessary):

Columns were read for readability and naming convention. Minor layout adjustments were offered where necessary (e.g., in order to replace spaces with underscores) to convert them into Python-syntax-friendly.

**Exploratory data analysis (EDA):**

The independent variables and the target variable (Diabetes\_Prevalence) were analyzed to uncover patterns, detect outliers, and guide decision-making for the modeling phase.

1. Univariate Analysis:

Independent variable was checked with:

• Histograms and KDE plots in a futile attempt to try out distribution of numeric features.

• Boxplots to separate occurrence of outliers and rough estimation of spread of values.

• Countplots where needed to look at frequency of values.

2. Bivariate Analysis:

Correlations between every independent variable and target were explored using:

• Singleton scatter plots to observe pattern of correlation.

• A correlation heatmap to identify multicollinearity and pairwise.

3. summary Statistics:

The descriptive stats such as mean, median, standard deviation, and skewness were calculated in order to provide an explanation of the behavior of the data and symmetry.

4. visualizations:

Minimum five different plots were utilized, which are:

• Histogram

• KDE Plot

• Boxplot

• Scatter plot

• Correlation Heatmap

5. Insights:

• We observe a clear positive trend between Avg\_Daily\_Sugar\_Intake and Diabetes\_Prevalence.

• Some indicators such as Sugar\_Import\_Export\_Ratio had lower correlations.

• Distribution of data appeared to be generally normal with some skewness in some features.

This EDA process meets the project requirement of analyzing at least six variables with the use of univariate and bivariate approaches and utilizing at least five types of visualizations, according to the project criteria.

**Feature Engineering and Feature Selection:**

This stage emphasized improvement of the performance of the model by constructing strong input variables and selecting the most appropriate relevant features to train on.

Secondly, feature engineering was also on the table with the aim of looking for potentialities of converting or creating new features out of given variables. But because the dataset was simple and already preprocessed, no explicit addition of new features took place. The focus was actually more inclined towards determining the contribution of current features towards the prediction of Diabetes\_Prevalence.

In feature selection, the SelectKBest function was applied with the chi-squared (chi²) statistical test. The technique ranks features based on their correlation with the target variable. The technique is a filtering technique, which examines each feature separately and retains those that have the highest statistical correlation with the target.

After SelectKBest application, the best-performing features were selected and employed for model training. This ensured that reduced model complexity with improved performance without permitting overfitting by eliminating less informative variables was attained.

The use of strict feature selection along with systematic input data ensured that subsequent modeling would be efficient and effective.

**Model Selection :**

In order to create a good predictive model for Diabetes\_Prevalence, various machine learning models were tried and compared. It was the intention to see which one performed optimally with the dataset.   
  
1. Model Selection  
  
The following three regression models were tried:  
• Linear Regression: Basic model as a baseline for comparison.  
• K-Nearest Neighbors (KNN): Non-parametric model that makes predictions based on the mean of the nearest neighbors.  
• Random Forest Regressor: Ensemble technique with an ensemble of a large number of decision trees for improving accuracy without overfitting.  
  
All the models were trained on the same preprocessed data and their performances were compared on the respective metrics (see below section).  
  
2. Model Tuning  
  
In order to further optimize performance, Random Forest model was hyperparameterized with GridSearchCV. GridSearchCV attempts all possible values of parameters in a range on cross-validation. The optimal parameters are:  
• n\_estimators: Number of trees  
• max\_depth: Depth of trees  
  
GridSearchCV discovers the best parameter combination having the maximum predicting power without overfitting.  
  
3. Results  
  
Random Forest algorithm was the best among all algorithms tested with regard. It was more accurate and always made better predictions when optimized and therefore was the production release choice.  
  
It made sure that the resulting model could not only be accurate, but also generalizable as well as efficient.

**Model Evaluation and Validation:**

Upon calibration and training of the models, their ability to make correct predictions for Diabetes\_Prevalence was then measured. Both validation techniques and measures of performance were used to determine the model's reliability and generalizability.  
  
1.Measures of Performance  
  
Measuring the performance of the models quantitatively, the following measures of regression were utilized:  
• R-squared (R²): The proportion of variance in the target variable that is explained by the independent variables. The larger the R², the more accurate the fit.  
• Mean Absolute Error (MAE): Defines the extent to which, on average, the predicted value differs from the actual value. Smaller MAE means better prediction.  
• Mean Squared Error (MSE): Puts more emphasis on large errors by squaring the differences. Lower is as good as MAE.  
  
These measures were utilized for comparison of performance of the three models. Random Forest Regressor provided the best combination of accuracy and decrease in error by utilizing the above measures.  
  
2. Validation  
  
In order to prevent overfitting and ensure that the model performs well with new unseen data, cross-validation was utilized while tuning. Exactly, GridSearchCV executed k-fold cross-validation because it was looking for optimal parameters. This minimized the chances of overfitting and provided a better estimate of the performance of the model.  
  
3. Performance Summary  
  
The trained Random Forest model also recorded high R² values and comparatively low MAE and MSE values. These are the indicators that the model can effectively predict the prevalence of diabetes from the characteristics of sugar consumption.

**Model Deployment:**

To bring the predictive model to practice and make it handy, the predictive model was implemented as a web application using the Flask framework. Deployment was an iterative process to transform the trained machine learning model into a dynamic web application.

The simple but effective user interface was built using an HTML form (index.html) where users can input four most significant variables:

• Average Daily Sugar Intake

• Per Capita Sugar Consumption

• Total Sugar Consumption

• Sugar Import/Export Ratio

When the form is submitted by the user, backend Flask application (app.py) stores the form values submitted, processes them, and loads saved trained model pickled from the saved file (diabetes\_model.pkl). The model calculates the predicted Diabetes Prevalence and returns it, which is ultimately displayed on the web page.

This deployment shows how a data science model can be transformed into a true, functional application from which the masses of non-technical people can benefit from the analysis. It also shows the end-to-end data science process—from data preprocessing to training a model and, finally, real-time prediction via a web application.