```sas
/*****************************************
 * Predictive Modeling for Diabetes Detection: A Comprehensive Approach Using SAS
 *****************************************/

/*****************************************
 * Data Import and Initial Overview
 *****************************************/

/* Import the diabetes dataset */
PROC IMPORT DATAFILE="/home/u64112808/sasuser.v94/Diabetes Prediction Project/diabetes.csv"
    OUT=diabetes_data
    DBMS=CSV
    REPLACE;
    GETNAMES=YES;
RUN;

/* Display dataset structure and metadata */
TITLE "Imported Diabetes Dataset Overview";
PROC CONTENTS DATA=diabetes_data;
RUN;
TITLE;

/* Display metadata */
TITLE "Metadata of Diabetes Dataset";
PROC CONTENTS DATA=diabetes_data;
RUN;
TITLE;

/* Display the first 10 rows of the dataset */
TITLE "Sample of the First 10 Rows in the Dataset";
PROC PRINT DATA=diabetes_data(OBS=10);
RUN;
TITLE;

/*****************************************
 * Checking and Handling Missing Values
 *****************************************/

/* Summary of missing values and basic statistics */
TITLE "Summary of Missing Values and Basic Statistics";
PROC MEANS DATA=diabetes_data N NMISS;
RUN;
TITLE;

/* Frequency distribution of the target variable */
TITLE "Frequency Distribution of Outcome Variable";
PROC FREQ DATA=diabetes_data;
    TABLES Outcome / MISSING;
RUN;
TITLE;

/* Check for invalid zeros in key variables */
TITLE "Checking Missing and Invalid Values in Key Variables";
PROC MEANS DATA=diabetes_data N NMISS MIN MAX;
    VAR Glucose BloodPressure SkinThickness Insulin BMI;
RUN;
TITLE;

/* Replace biologically invalid zeros with missing values */
DATA diabetes_clean;
    SET diabetes_data;
    IF Glucose = 0 THEN Glucose = .;
    IF BloodPressure = 0 THEN BloodPressure = .;
    IF SkinThickness = 0 THEN SkinThickness = .;
    IF Insulin = 0 THEN Insulin = .;
    IF BMI = 0 THEN BMI = .;
RUN;

/* Validate the dataset after replacing invalid zeros */
TITLE "Summary After Replacing Invalid Zeros with Missing Values";
PROC MEANS DATA=diabetes_clean N NMISS MIN MAX;
    VAR Glucose BloodPressure SkinThickness Insulin BMI;
RUN;
TITLE;

/* Distribution of missing values after cleaning */
TITLE "Checking Distribution of Missing Values After Cleaning";
PROC MEANS DATA=diabetes_clean N NMISS;
RUN;
TITLE;

/*****************************************
 * Imputation of Missing Values
 *****************************************/

/* Calculate means for missing value imputation */
PROC MEANS DATA=diabetes_clean NOPRINT;
```

```
    VAR Glucose BloodPressure SkinThickness Insulin BMI;
    OUTPUT OUT=mean_values
        MEAN=Mean_Glucose Mean_BP Mean_ST Mean_Insulin Mean_BMI;
RUN;

/* Impute missing values using calculated means */
DATA diabetes_imputed;
    SET diabetes_clean;
    IF _N_ = 1 THEN SET mean_values;
    IF MISSING(Glucose) THEN Glucose = Mean_Glucose;
    IF MISSING(BloodPressure) THEN BloodPressure = Mean_BP;
    IF MISSING(SkinThickness) THEN SkinThickness = Mean_ST;
    IF MISSING(Insulin) THEN Insulin = Mean_Insulin;
    IF MISSING(BMI) THEN BMI = Mean_BMI;
RUN;

/* Validate the dataset after imputation */
TITLE "Summary After Correcting Imputation Logic";
PROC MEANS DATA=diabetes_imputed N NMISS MIN MAX;
RUN;
TITLE;

/* Display the first 10 rows of the final dataset */
TITLE "Final Cleaned and Preprocessed Dataset After Correct Imputation";
PROC PRINT DATA=diabetes_imputed(OBS=10);
RUN;
TITLE;

/****************************************
 * Descriptive Statistics
 ****************************************/

/* Generate summary statistics for all numeric variables */
TITLE "Descriptive Statistics for Key Variables";
PROC MEANS DATA=diabetes_imputed N MEAN STD MIN MAX;
    VAR Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age;
RUN;
TITLE;

/****************************************
 * Distribution Analysis
 ****************************************/

/* Analyze the distribution of glucose levels */
TITLE "Distribution of Glucose Levels";
PROC SGPLOT DATA=diabetes_imputed;
    HISTOGRAM Glucose / BINWIDTH=10;
    DENSITY Glucose;
    XAXIS LABEL="Glucose Level";
    YAXIS LABEL="Frequency";
RUN;

/* Analyze the distribution of BMI */
TITLE "Distribution of BMI";
PROC SGPLOT DATA=diabetes_imputed;
    HISTOGRAM BMI / BINWIDTH=2;
    DENSITY BMI;
    XAXIS LABEL="Body Mass Index (BMI)";
    YAXIS LABEL="Frequency";
RUN;

/* Analyze the distribution of age */
TITLE "Distribution of Age";
PROC SGPLOT DATA=diabetes_imputed;
    HISTOGRAM Age / BINWIDTH=5;
    DENSITY Age;
    XAXIS LABEL="Age (Years)";
    YAXIS LABEL="Frequency";
RUN;
TITLE;

/****************************************
 * Correlation Analysis
 ****************************************/

/* Compute correlations between key variables */
TITLE "Correlation Analysis of Key Variables";
PROC CORR DATA=diabetes_imputed PLOTS=MATRIX;
    VAR Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age;
RUN;
TITLE;

/****************************************
 * Target Variable Analysis
 ****************************************/

/* Analyze the distribution of the target variable */
TITLE "Distribution of Outcome (Diabetes vs. Non-Diabetes)";
```

```sas
PROC FREQ DATA=diabetes_imputed;
    TABLES Outcome / PLOTS=FREQPLOT;
RUN;

/* Boxplot analysis for glucose by outcome */
TITLE "Boxplot of Glucose by Outcome";
PROC SGPLOT DATA=diabetes_imputed;
    VBOX Glucose / CATEGORY=Outcome;
    XAXIS LABEL="Outcome (0: No Diabetes, 1: Diabetes)";
    YAXIS LABEL="Glucose Level";
RUN;

/* Boxplot analysis for BMI by outcome */
TITLE "Boxplot of BMI by Outcome";
PROC SGPLOT DATA=diabetes_imputed;
    VBOX BMI / CATEGORY=Outcome;
    XAXIS LABEL="Outcome (0: No Diabetes, 1: Diabetes)";
    YAXIS LABEL="BMI";
RUN;

/* Boxplot analysis for age by outcome */
TITLE "Boxplot of Age by Outcome";
PROC SGPLOT DATA=diabetes_imputed;
    VBOX Age / CATEGORY=Outcome;
    XAXIS LABEL="Outcome (0: No Diabetes, 1: Diabetes)";
    YAXIS LABEL="Age (Years)";
RUN;
TITLE;

/****************************************
 * Feature Scaling
 ****************************************/

/* Calculate mean and standard deviation for Glucose, BMI, and Age */
PROC MEANS DATA=diabetes_imputed NOPRINT;
    VAR Glucose BMI Age;
    OUTPUT OUT=stats MEAN=Mean_Glucose Mean_BMI Mean_Age
                     STD=Std_Glucose Std_BMI Std_Age;
RUN;

/* Standardize Glucose, BMI, and Age using calculated means and standard deviations */
DATA diabetes_scaled;
    SET diabetes_imputed;
    IF _N_ = 1 THEN SET stats;
    Z_Glucose = (Glucose - Mean_Glucose) / Std_Glucose;
    Z_BMI = (BMI - Mean_BMI) / Std_BMI;
    Z_Age = (Age - Mean_Age) / Std_Age;
RUN;

/* Verify the scaled variables */
TITLE "Summary of Scaled Features (Z_Glucose, Z_BMI, Z_Age)";
PROC MEANS DATA=diabetes_scaled N MEAN STD MIN MAX;
    VAR Z_Glucose Z_BMI Z_Age;
RUN;
TITLE;

/****************************************
 * Feature Engineering
 ****************************************/

/* Add interaction terms and categorize BMI */
DATA diabetes_engineered;
    SET diabetes_scaled;

    /* Interaction term: Glucose and BMI */
    Interaction_Glucose_BMI = Z_Glucose * Z_BMI;

    /* BMI categories */
    IF BMI < 18.5 THEN BMI_Category = "Underweight";
    ELSE IF BMI >= 18.5 AND BMI < 25 THEN BMI_Category = "Normal";
    ELSE IF BMI >= 25 AND BMI < 30 THEN BMI_Category = "Overweight";
    ELSE BMI_Category = "Obese";
RUN;

/* Verify engineered features */
TITLE "Summary of Engineered Features (Interaction_Glucose_BMI and BMI_Category)";
PROC MEANS DATA=diabetes_engineered N MEAN STD MIN MAX;
    VAR Interaction_Glucose_BMI;
RUN;

PROC FREQ DATA=diabetes_engineered;
    TABLES BMI_Category;
RUN;
TITLE;

/****************************************
 * Logistic Regression Model
 ****************************************/
```

```
/* Logistic regression to predict Outcome (Diabetes) */
TITLE "Logistic Regression Model: Predicting Diabetes Outcome";
PROC LOGISTIC DATA=diabetes_engineered DESCENDING;
    CLASS BMI_Category (REF="Normal"); /* Specify BMI_Category as a CLASS variable */
    MODEL Outcome = Z_Glucose Z_BMI Z_Age Interaction_Glucose_BMI BMI_Category / SELECTION=STEPWISE;
    OUTPUT OUT=logistic_results PREDICTED=Predicted_Prob;
RUN;
TITLE;


/*****************************************
 * Confusion Matrix and Performance Metrics
 *****************************************/
/* Create a binary prediction variable based on a 0.5 threshold */
DATA logistic_results;
    SET logistic_results;
    Predicted_Class = (Predicted_Prob >= 0.5); /* 1 = Diabetes, 0 = No Diabetes */
RUN;

/* Generate confusion matrix */
TITLE "Confusion Matrix for Logistic Regression Predictions";
PROC FREQ DATA=logistic_results;
    TABLES Outcome * Predicted_Class / CHISQ NOROW NOCOL NOPERCENT;
RUN;
TITLE;

/*****************************************
 * Compute Performance Metrics
 *****************************************/

/* Summarize confusion matrix values */
PROC SQL;
    SELECT
        SUM(CASE WHEN Outcome = 1 AND Predicted_Class = 1 THEN 1 ELSE 0 END) AS TP, /* True Positives */
        SUM(CASE WHEN Outcome = 0 AND Predicted_Class = 0 THEN 1 ELSE 0 END) AS TN, /* True Negatives */
        SUM(CASE WHEN Outcome = 0 AND Predicted_Class = 1 THEN 1 ELSE 0 END) AS FP, /* False Positives */
        SUM(CASE WHEN Outcome = 1 AND Predicted_Class = 0 THEN 1 ELSE 0 END) AS FN /* False Negatives */
    INTO :TP, :TN, :FP, :FN
    FROM logistic_results;
QUIT;

/* Calculate and display performance metrics */
DATA performance_metrics;
    TP = &TP;
    TN = &TN;
    FP = &FP;
    FN = &FN;

    Accuracy = (TP + TN) / (TP + TN + FP + FN);
    Precision = TP / (TP + FP);
    Recall = TP / (TP + FN);
    Specificity = TN / (TN + FP);
    F1_Score = 2 * (Precision * Recall) / (Precision + Recall);

    OUTPUT;
RUN;

/* Display performance metrics */
TITLE "Performance Metrics for Logistic Regression Model";
PROC PRINT DATA=performance_metrics;
    VAR TP TN FP FN Accuracy Precision Recall Specificity F1_Score;
    FORMAT Accuracy Precision Recall Specificity F1_Score 8.3; /* Format metrics for readability */
RUN;
TITLE;

/*****************************************
 * ROC Curve and AUC
 *****************************************/

/* Generate ROC curve and calculate AUC */
TITLE "ROC Curve and AUC for Logistic Regression";
PROC LOGISTIC DATA=diabetes_engineered PLOTS(ONLY)=ROC;
    CLASS BMI_Category (REF="Normal");
    MODEL Outcome = Z_Glucose Z_BMI Z_Age Interaction_Glucose_BMI BMI_Category;
    OUTPUT OUT=roc_results PREDICTED=Predicted_Prob;
RUN;
TITLE;


/*****************************************
 * Refined Logistic Regression Model
 *****************************************/

TITLE "Refined Logistic Regression Model: Excluding Non-Significant Interaction Term";
PROC LOGISTIC DATA=diabetes_engineered DESCENDING;
    CLASS BMI_Category (REF="Normal");
    MODEL Outcome = Z_Glucose Z_BMI Z_Age BMI_Category / SELECTION=STEPWISE;
```

```
    OUTPUT OUT=refined_logistic_results PREDICTED=Predicted_Prob;
RUN;
TITLE;

/*****************************************
 * Performance Metrics for Refined Logistic Regression
 *****************************************/

/* Create a binary prediction variable based on a 0.5 threshold */
DATA refined_logistic_results;
    SET refined_logistic_results;
    Predicted_Class = (Predicted_Prob >= 0.5); /* 1 = Diabetes, 0 = No Diabetes */
RUN;

/* Generate confusion matrix */
TITLE "Confusion Matrix for Refined Logistic Regression Predictions";
PROC FREQ DATA=refined_logistic_results;
    TABLES Outcome * Predicted_Class / CHISQ NOROW NOCOL NOPERCENT;
RUN;
TITLE;

/* Summarize confusion matrix values */
PROC SQL;
    SELECT
        SUM(CASE WHEN Outcome = 1 AND Predicted_Class = 1 THEN 1 ELSE 0 END) AS TP, /* True Positives */
        SUM(CASE WHEN Outcome = 0 AND Predicted_Class = 0 THEN 1 ELSE 0 END) AS TN, /* True Negatives */
        SUM(CASE WHEN Outcome = 0 AND Predicted_Class = 1 THEN 1 ELSE 0 END) AS FP, /* False Positives */
        SUM(CASE WHEN Outcome = 1 AND Predicted_Class = 0 THEN 1 ELSE 0 END) AS FN /* False Negatives */
    INTO :TP, :TN, :FP, :FN
    FROM refined_logistic_results;
QUIT;

/* Calculate and display performance metrics */
DATA refined_performance_metrics;
    TP = &TP;
    TN = &TN;
    FP = &FP;
    FN = &FN;

    Accuracy = (TP + TN) / (TP + TN + FP + FN);
    Precision = TP / (TP + FP);
    Recall = TP / (TP + FN);
    Specificity = TN / (TN + FP);
    F1_Score = 2 * (Precision * Recall) / (Precision + Recall);

    OUTPUT;
RUN;

/*****************************************
 * Import the Test Dataset
 *****************************************/
PROC IMPORT DATAFILE="/home/u64112808/sasuser.v94/Diabetes Prediction Project/test_data.csv"
    OUT=new_data
    DBMS=CSV
    REPLACE;
    GETNAMES=YES;
RUN;

/* Display the structure of the imported data */
TITLE "Structure of the Imported Test Dataset";
PROC CONTENTS DATA=new_data;
RUN;
TITLE;


/*****************************************
 * Prepare the Test Dataset
 *****************************************/
DATA new_data_processed;
    SET new_data;
    /* Standardize variables using actual training dataset means and standard deviations */
    Z_Glucose = (Glucose - 121.69) / 30.44;
    Z_BMI = (BMI - 32.46) / 6.88;
    Z_Age = (Age - 33.24) / 11.76;
    /* Assign BMI categories */
    IF BMI < 18.5 THEN BMI_Category = "Underweight";
    ELSE IF BMI >= 18.5 AND BMI < 25 THEN BMI_Category = "Normal";
    ELSE IF BMI >= 25 AND BMI < 30 THEN BMI_Category = "Overweight";
    ELSE BMI_Category = "Obese";
RUN;

/* Verify the processed data */
TITLE "Processed Test Dataset";
PROC PRINT DATA=new_data_processed(OBS=10);
RUN;
TITLE;

/*****************************************
```

```
 * Score Processed Test Dataset
 ****************************************/
PROC LOGISTIC INMODEL=refined_model;
    SCORE DATA=new_data_processed OUT=new_data_predictions;
RUN;


/* Check if scoring was successful */
TITLE "Contents of Scored Data";
PROC CONTENTS DATA=new_data_predictions;
RUN;
TITLE;

/****************************************
 * Add Predicted Classes
 ****************************************/
DATA new_data_predictions;
    SET new_data_predictions;
    Predicted_Prob = P_1; /* Map predicted probability for Outcome=1 */
    Predicted_Class = (Predicted_Prob >= 0.5); /* Binary classification: 1 = Diabetes, 0 = No Diabetes */
RUN;


/****************************************
 * View Predictions
 ****************************************/
TITLE "Predictions for Processed Test Dataset";
PROC PRINT DATA=new_data_predictions(OBS=10);
    VAR Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome Predicted_Prob Predict
RUN;
TITLE;

/****************************************
 * Export Predictions to CSV
 ****************************************/
PROC EXPORT DATA=new_data_predictions
    OUTFILE="/home/u64112808/sasuser.v94/Diabetes Prediction Project/predicted_data.csv"
    DBMS=CSV
    REPLACE;
RUN;

/* Confirmation */
TITLE "Predictions Exported to CSV";
RUN;
```