```
/*Step 1: Import Data*/
/*First, we need to import the German credit data into SAS.*/

FILENAME REFFILE '/home/u64024530/sasuser.v94/FINAL PROJECT_CIND119/german_credit.csv';

PROC IMPORT DATAFILE=REFFILE DBMS=CSV OUT=WORK.GERMAN_CREDIT;
    GETNAMES=YES;
RUN;

/*Step 2: Explore the Data*/
/*Understand the structure and contents of our data by running some exploratory commands.*/
/*2.1: View the First Few Rows*/
proc print data=GERMAN_CREDIT(obs=10);
run;

/*2.2: Get a summary of the Dataset */
PROC CONTENTS DATA=GERMAN_CREDIT;
RUN;

/*2.3: Generate descriptive statistics for numerical variables */
PROC MEANS DATA=WORK.GERMAN_CREDIT N MEAN STD MIN MAX;
RUN;

/*Step 2.4:Frequency Distribution*/
proc freq data=GERMAN_CREDIT;
    tables Creditability    AccountBalance  DurationofCredit    PaymentStatusofPreviousCredit   Purpose CreditAmount    ValueSavingsandStocks   Lengthofcurrentemploym
    run;
/*Step 2.5: Visualize the data using histograms for numerical variables */
PROC UNIVARIATE DATA=WORK.GERMAN_CREDIT;
    HISTOGRAM _NUMERIC_;
RUN;

/*Step 3: Data Preparation*/
/*Clean our data by handling missing values, encoding categorical variables, and partitioning the data.*/
/*Step 3.1: Identify and Count Missing Values */
PROC MEANS DATA=WORK.GERMAN_CREDIT N NMISS;
    VAR _NUMERIC_;
RUN;

/*Step 3.2:Handle Missing Values*/
/*To handle missing values, you can choose either to remove them or impute them. In this example, we'll simply identify and count them. If needed, you can use imputat
/* Handle categorical missing values with PROC FREQ */
proc freq data=GERMAN_CREDIT;
    tables Creditability AccountBalance PaymentStatusofPreviousCredit Purpose
                          ValueSavingsandStocks Lengthofcurrentemployment Instalmentpercent
                          SexandMaritalStatus Guarantors DurationinCurrentaddress
                          Mostvaluableavailableasset ConcurrentCredits Typeofapartment
                          NoofCreditsatthisBank Occupation Noofdependents Telephone ForeignWorker
/ missing;
run;

/* Example of Mean Imputation for Specific Numeric Variables */

/* First, examine which of these variables are numeric and have missing values */
PROC MEANS DATA=WORK.GERMAN_CREDIT N NMISS MEAN;
    VAR Creditability   AccountBalance  DurationofCredit    PaymentStatusofPreviousCredit   Purpose CreditAmount    ValueSavingsandStocks   Lengthofcurrentemployment
RUN;
```

```
/*Step 3.3: Apply mean imputation for variables with missing values */
DATA WORK.GERMAN_CREDIT_IMPUTED;
    SET WORK.GERMAN_CREDIT;

    /* Impute each numeric variable individually */
    IF MISSING(Creditability) THEN Creditability = MEAN(Creditability);
    IF MISSING(AccountBalance) THEN AccountBalance = MEAN(AccountBalance);
    IF MISSING(PaymentStatusofPreviousCredit) THEN PaymentStatusofPreviousCredit = MEAN(PaymentStatusofPreviousCredit);
    IF MISSING(ValueSavingsandStocks) THEN ValueSavingsandStocks = MEAN(ValueSavingsandStocks);
    IF MISSING(Instalmentpercent) THEN Instalmentpercent = MEAN(Instalmentpercent);
    IF MISSING(NoofCreditsatthisBank) THEN NoofCreditsatthisBank = MEAN(NoofCreditsatthisBank);
    IF MISSING(Noofdependents) THEN Noofdependents = MEAN(Noofdependents);
RUN;
/*Step 3.4: Encode Categorical Variables
Convert categorical variables into a numerical format, often using one-hot encoding or dummy variables.*/
/* One-hot encode categorical variables */
/* Encoding Categorical Variables using PROC GLMMOD and PROC GLMSELECT */
PROC GLMMOD DATA=WORK.GERMAN_CREDIT OUTDESIGN=WORK.GERMAN_CREDIT_ENCODED;
    CLASS AccountBalance PaymentStatusofPreviousCredit Purpose
                          ValueSavingsandStocks Lengthofcurrentemployment Instalmentpercent
                          SexandMaritalStatus Guarantors DurationinCurrentaddress
                          Mostvaluableavailableasset ConcurrentCredits Typeofapartment
                          NoofCreditsatthisBank Occupation Noofdependents Telephone ForeignWorker;

    MODEL Creditability = AccountBalance PaymentStatusofPreviousCredit Purpose CreditAmount
                          ValueSavingsandStocks Lengthofcurrentemployment Instalmentpercent
                          SexandMaritalStatus Guarantors DurationinCurrentaddress
                          Mostvaluableavailableasset ConcurrentCredits Typeofapartment
                          NoofCreditsatthisBank Occupation Noofdependents Telephone ForeignWorker;
RUN;

proc glmselect data=GERMAN_CREDIT outdesign=german_credit_cleaned;
     CLASS AccountBalance PaymentStatusofPreviousCredit Purpose
                          ValueSavingsandStocks Lengthofcurrentemployment Instalmentpercent
                          SexandMaritalStatus Guarantors DurationinCurrentaddress
                          Mostvaluableavailableasset ConcurrentCredits Typeofapartment
                          NoofCreditsatthisBank Occupation Noofdependents Telephone ForeignWorker;

    MODEL Creditability = AccountBalance PaymentStatusofPreviousCredit Purpose
                          ValueSavingsandStocks Lengthofcurrentemployment Instalmentpercent
                          SexandMaritalStatus Guarantors DurationinCurrentaddress
                          Mostvaluableavailableasset Age ConcurrentCredits Typeofapartment
                          NoofCreditsatthisBank Occupation Noofdependents Telephone ForeignWorker/ selection=none;
run;

/* Step 3.5:Partition the Data*/
/*Splitting Data into Training and Testing */
data german_credit_cleaned;
    set GERMAN_CREDIT;
run;

proc surveyselect data=german_credit_cleaned out=german_credit_train samprate=0.7/* 70% for training */ seed=12345/* For reproducibility */outall;
run;

/* Set training and testing datasets */
DATA TRAIN TEST;
    SET WORK.GERMAN_CREDIT_PART;
    IF SELECTED THEN OUTPUT TRAIN;
    ELSE OUTPUT TEST;
RUN;
data train test;
    set german_credit_train;

    if selected then
        output train;
    else
        output test;
run;

proc contents data=GERMAN_CREDIT_TRAIN; run;

/*Step 4: Build the Regression Model*/
/*Use PROC LOGISTIC for logistic regression, often suitable for risk assessment models.*/
/* Run logistic regression */
proc logistic data=german_credit_train outmodel=GERMAN_CREDIT_MODEL;
    class AccountBalance PaymentStatusofPreviousCredit  Purpose ValueSavingsandStocks   Instalmentpercent   Guarantors  Mostvaluableavailableasset ConcurrentCredits
    model Creditability(event='1') = AccountBalance PaymentStatusofPreviousCredit   Purpose ValueSavingsandStocks   Instalmentpercent Guarantors    Mostvaluableavaila
run;

/*Step 5: Validate Logistic Model
Evaluate the model using the test dataset to verify its performance.*/
proc logistic inmodel=GERMAN_CREDIT_MODEL;
    score data=GERMAN_CREDIT_TRAIN out=predictions;
run;
/*5.1. ROC Curve and AUC*/
/*The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system. The Area Under the Curv

proc logistic data=GERMAN_CREDIT_TRAIN;
        model Creditability(event='1') = AccountBalance PaymentStatusofPreviousCredit   Purpose ValueSavingsandStocks   Instalmentpercent Guarantors    Mostvaluableav
    roc 'ROC Curve';
run;

/*5.2. Confusion Matrix*/
/*A confusion matrix provides a summary of prediction results on a classification problem. It shows the number of correct and incorrect predictions broken down by eac
proc freq data=german_credit_train;
    tables Creditability*AccountBalance PaymentStatusofPreviousCredit   Purpose ValueSavingsandStocks   Instalmentpercent Guarantors    Mostvaluableavailableasset Con
run;

proc freq data=german_credit_train;
    tables Creditability*AccountBalance PaymentStatusofPreviousCredit   Purpose ValueSavingsandStocks   Instalmentpercent Guarantors    Mostvaluableavailableasset Con
run;


/*6.3. Cross-Validation*/
/*Cross-validation is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. In SAS, you can perform cross-va
proc glmselect data=german_credit_train;
    class AccountBalance PaymentStatusofPreviousCredit Purpose;
    model Creditability = AccountBalance PaymentStatusofPreviousCredit Purpose CreditAmount
                          ValueSavingsandStocks Lengthofcurrentemployment Instalmentpercent
                          SexandMaritalStatus Guarantors DurationinCurrentaddress
                          Mostvaluableavailableasset Age ConcurrentCredits Typeofapartment
                          NoofCreditsatthisBank Occupation Noofdependents Telephone ForeignWorker
                          / selection=stepwise(select=SL) details=all;
    partition fraction(validate=0.3);
run;
```

```
/*Step 7:Built Decision Tree*/
proc hpsplit data=german_credit_train;
    class AccountBalance PaymentStatusofPreviousCredit  Purpose ValueSavingsandStocks   Instalmentpercent   Guarantors  Mostvaluableavailableasset ConcurrentCredits
    model Creditability = AccountBalance PaymentStatusofPreviousCredit  Purpose ValueSavingsandStocks   Instalmentpercent Guarantors    Mostvaluableavailableasset Con
    grow gini; /* Use Gini index for splitting */
    prune costcomplexity; /* Prune the tree using cost complexity */
run;
/*Step 7.1:Validate the Decision Tree Model*/
/*To validate the model, you can use a test dataset to assess its performance.*/
proc hpsplit data=test;
       class AccountBalance PaymentStatusofPreviousCredit   Purpose ValueSavingsandStocks   Instalmentpercent   Guarantors  Mostvaluableavailableasset ConcurrentCredi
       model Creditability = AccountBalance PaymentStatusofPreviousCredit   Purpose ValueSavingsandStocks   Instalmentpercent Guarantors    Mostvaluableavailableasset

    code file='tree_code.sas';
run;
```