Project Guideline

The columns in this dataset are:

* **Loan\_ID:** A unique identifier for each loan.
* **Gender:** The gender of the loan applicant (Male or Female).
* **Married:** Whether the applicant is married (Yes or No).
* **Dependents:** The number of dependents the applicant has.
* **Education:** The highest level of education attained by the applicant (Graduate or Under Graduate).
* **Self\_Employed:** Whether the applicant is self-employed (Yes or No).
* **ApplicantIncome:** The annual income of the applicant.
* **CoapplicantIncome:** The annual income of the coapplicant (if applicable).
* **LoanAmount:** The loan amount requested, in thousands of dollars.
* **Loan\_Amount\_Term:** The duration of the loan, in months.
* **Credit\_History:** Whether the applicant's credit history meets the lender's guidelines (Yes or No).
* **Property\_Area:** The location of the property for which the loan is being requested (Urban, Semi Urban, or Rural).
* **Loan\_Status:** The final decision on whether the loan is approved or not (Yes or No). This is typically considered the target variable in a machine learning model that predicts loan approval.

**1. Import necessary libraries:**

* pandas is used for data manipulation and analysis.
* matplotlib.pyplot is used for creating visualizations.
* seaborn is used for creating statistical graphics.
* sklearn.model\_selection is used for splitting data into training and testing sets.
* sklearn.linear\_model is used for training a logistic regression model.
* sklearn.metrics is used for evaluating the model's performance.

**2. Load the data:**

* The data is loaded from a CSV file named loan\_prediction.csv.
* The data is stored in a pandas DataFrame named data.

**3. Explore the data:**

* The first 5 rows of the data are displayed using the head() method.
* The last 5 rows of the data are displayed using the tail() method.
* The shape of the data is displayed using the shape attribute.
* The data types of each column are displayed using the info() method.
* The number of missing values in each column are displayed using the isnull() method.

**4. Handle missing values:**

* The missing values in the Gender, Married, Dependents, Self\_Employed, LoanAmount, and Loan\_Amount\_Term columns are filled with the mode of the column.
* The missing values in the Credit\_History column are filled with the most frequent value (which is 1).

**5. Split the data into training and testing sets:**

* The data is split into training and testing sets using the train\_test\_split() function.
* The training set is used to train the model, and the testing set is used to evaluate the model's performance.

**6. Train the logistic regression model:**

* A logistic regression model is trained on the training set using the LogisticRegression() class.
* The model is trained using the fit() method.

**7. Evaluate the model's performance:**

* The model's performance is evaluated on the testing set using the accuracy\_score() function.
* The accuracy score is printed to the console.

**8. Save the model:**

* The trained model is saved to a file using the joblib.dump() function.

**9. Load the saved model:**

* The saved model is loaded from a file using the joblib.load() function.

**10. Make predictions:**

* The model is used to make predictions on new data using the predict() method.

**Here are some additional details about the steps taken in the project:**

* The Gender, Married, Dependents, Self\_Employed, LoanAmount, and Loan\_Amount\_Term columns are categorical variables, so they are filled with the mode of the column.
* The Credit\_History column is a binary variable, so it is filled with the most frequent value (which is 1).
* The train\_test\_split() function is used to split the data into training and testing sets. The test\_size parameter is set to 0.2, which means that 20% of the data is used for testing and 80% of the data is used for training.
* The LogisticRegression() class is used to train a logistic regression model. The C parameter is set to 1.0, which is the regularization parameter.
* The accuracy\_score() function is used to calculate the accuracy of the model. The accuracy is the number of correct predictions divided by the total number of predictions.
* The joblib.dump() function is used to save the model to a file. The joblib.load() function is used to load the model from a file.
* The predict() method is used to make predictions on new data. The input to the predict() method is a pandas DataFrame that contains the same columns as the training data.

### **Steps Followed**

1. **Data Loading and Initial Exploration**
   * Imported necessary libraries, including pandas and google.colab.drive.
   * Loaded the dataset from Google Drive.
   * Displayed the first and last 5 rows of the dataset.
   * Checked the shape of the dataset.
   * Obtained information about the dataset, including the number of rows, columns, and data types of each column.
   * Identified columns with missing values and calculated the percentage of missing values.
2. **Data Cleaning**
   * Dropped the Loan\_ID column, as it was not required.
   * Dropped columns with missing values less than 5%.
   * Filled remaining missing values with appropriate strategies (mean, median, or mode).
3. **Data Preprocessing**
   * Converted categorical variables into numerical values using label encoding.
   * Checked for and addressed multicollinearity using correlation matrix and VIF (Variance Inflation Factor).
4. **Model Building**
   * Split the data into training and testing sets.
   * Trained and evaluated several models, including Logistic Regression, Decision Tree Classifier, Random Forest Classifier, K-Nearest Neighbors, and Support Vector Classifier.
   * Used GridSearchCV for hyperparameter tuning of the best performing models.
5. **Model Evaluation**
   * Evaluated models using accuracy scores before and after hyperparameter tuning.
   * Chose the Random Forest Classifier based on performance metrics.
6. **Model Deployment**
   * Saved the trained model using joblib.
   * Loaded the saved model and made predictions on new data.
   * Displayed the loan approval status for each prediction.