RECIPE 1
1st Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`label_distribution` shows class proportions for train, val, test
Sampling	Yes	`train_test_split` with 50%-25%-25% split; random sampling
Outliers removal	Yes	Drop rows where `Num_Bank_Accounts > 30`, `< 0`, same for `Num_of_Loan`, `Num_Credit_Card`
Check for duplicates	Yes	`dataframe.duplicated().sum()` used to count duplicates
Imputation	Yes	`SimpleImputer(strategy='mean')` for numerical, `most_frequent` for categorical
Drop columns	Yes	Columns like `ID`, `SSN`, `Name`, `Monthly_Inhand_Salary`, `Customer_ID` dropped
Encoding	Yes	Used `LabelEncoder()` on categorical columns
Create new columns	No	No new columns created
Feature selection	Yes	Manual drop of `Monthly_Inhand_Salary` due to high correlation
Scaling / Standardisation	No	No standardization or normalization seen
Hyperparameter tuning	Yes	`GridSearchCV` for Logistic Regression and Random Forest

2nd Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Distribution of target `Credit_Score` is checked across train/val/test

Sampling type	Random	`train_test_split(..., shuffle=True)` used without stratification
Outliers removal	Yes	Removed based on logical thresholds for `Num_Bank_Accounts`, `Num_of_Loan`, `Num_Credit_Card`
Check for duplicates	Yes	`dataframe.duplicated().sum()` used
Imputation of missing values	Mixture of imputation techniques	Mean imputation (SimpleImputer) + mode imputation (groupby + transform + mode) + replace _ with NaN
Drop columns	Yes	Columns like `ID`, `SSN`, `Monthly_Inhand_Salary`, `Customer_ID`, etc. dropped without replacement
Encoding	Label Encoder	`LabelEncoder()` used for categorical columns
Create new columns	No	All derived columns (e.g. transform using groupby) are not counted as new columns per Prompt-2 & Prompt-3
Feature selection	Yes	`Monthly_Inhand_Salary` dropped after correlation analysis
Data scaling/standardisation	No	No scaling method (e.g. StandardScaler, MinMaxScaler) used
Hyperparameter tuning	Yes	`GridSearchCV` used for both Logistic Regression and Random Forest

3rd Prompt
Accuracy 10/11

Step	Used?	Details
Check for balanced data	Yes	Class distribution checked with `value_counts(normalize=True)`
Sampling type	Random	`train_test_split(..., shuffle=True)`

Outlier removal	Yes	Dropped based on logical conditions on `Num_Bank_Accounts`, `Num_Credit_Card`, etc.
Check for duplicates	Yes	Used `dataframe.duplicated().sum()`
Missing value imputation	Yes	Used `SimpleImputer` (mean + most_frequent)
Drop columns	Yes	Multiple drops: `ID`, `SSN`, `Monthly_Inhand_Salary`, etc.
Encoding	LE	Used `LabelEncoder()`
Create new columns	Yes	Group-wise imputation for mode values by `Customer_ID`
Feature selection	Yes	Correlation heatmap + manual removal
Standardization	No	No scaler used
Hyperparameter tuning	Yes	`GridSearchCV` used for LR and RF

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`value_counts(normalize=True)` used on target across train/val/test to assess balance.
Sampling type	Random	`train_test_split(..., shuffle=True)` without `stratify=`
Outliers removal	Yes	Values removed based on thresholds (e.g., `Num_Bank_Accounts > 30`, `< 0`, etc.)
Check for duplicates	Yes	`dataframe.duplicated().sum()` was used
Imputation of missing values	mixture of imputation techniques	Used `SimpleImputer` with `mean` for numeric, `most_frequent` for categorical, and dropped remaining rows
Drop columns	Yes	`'ID','Month','SSN','Name'` and others are dropped without replacement

Encoding	Label Encoder	All categorical columns encoded using `LabelEncoder()`
Create new columns	No	All transformations were replacements or derived (e.g., `.str.replace()`, `.transform()`)
Feature selection	Yes	Dropped `Monthly_Inhand_Salary` post correlation heatmap analysis
Data scaling/standardisation	No	No standard scaler, min-max scaler, or robust scaler used
Hyperparameter tuning	Yes	GridSearchCV used for both Logistic Regression and Random Forest

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Target label (`Credit_Score`) distribution checked using `value_counts(normalize=True)` across train/val/test.
Sampling type	Random	Train/val/test split done using `train_test_split` with `shuffle=True` but without stratify.
Outliers removal	Yes	Dropped rows where `Num_Bank_Accounts`, `Num_of_Loan`, `Num_Credit_Card` exceeded logical limits or were negative.
Check for duplicates	Yes	Used `dataframe.duplicated().sum()` to count duplicate records.
Imputation of missing values	mixture of imputation techniques	Numerical: mean imputation using `SimpleImputer`. Categorical: mode imputation and `dropna` for some columns.
Drop columns	Yes	`Monthly_Inhand_Salary`, `Customer_ID`, `ID`, etc., dropped

Encoding	Label Encoder	All categorical columns encoded using `LabelEncoder()`.
Create new columns	No	No evidence of entirely new columns added; transformations and imputations only.
Feature selection	Yes	Columns like `Monthly_Inhand_Salary` dropped based on correlation heatmap and EDA.
Data scaling/standardisation	No	No scaler (StandardScaler/MinMax/Robust) used on numerical columns.
Hyperparameter tuning	Yes	GridSearchCV applied for both Logistic Regression and Random Forest (with `param_grid`, `cv`, and `n_jobs=-1`).

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Class balance inspected via `value_counts(normalize=True)` for `Credit_Score` across overall/train/val/test (`label_distribution` table).
Sampling type	Random	Two-step split with `train_test_split(..., shuffle=True, random_state=42)` to create Train/Val/Test (no `stratify`).
Outliers removal	Yes	Rule-based row drops: negative values in `Num_Bank_Accounts`, `Num_of_Loan`; caps `Num_Bank_Accounts > 30`, `Num_Credit_Card > 30`.
Check for duplicates	Yes	Checked with `dataframe.duplicated().sum()`.
Imputation of missing values	Mixture of imputation techniques	Numeric means via `SimpleImputer(strategy='mean')` for several columns; categorical most-frequent via `SimpleImputer(strategy='most_frequent')`; plus row drops for remaining NAs in key categoricals.

Drop columns	Yes	Dropped `['ID','Month','SSN','Name']` early; later dropped `Monthly_Inhand_Salary` (collinearity) and `Customer_ID` (identifier), none reused.
Encoding	Label Encoder	In-place label encoding on all categorical columns via `LabelEncoder().fit_transform(...)`.
Create new columns	No	No genuinely new features for modelling (no one-hot expansion; only in-place recoding).
Feature selection	Yes	Correlation/EDA-driven drop of `Monthly_Inhand_Salary` after heatmap analysis (drop-after-visualisation counts as selection).
Data scaling/standardisation	No	No scaler (`StandardScaler`, `MinMaxScaler`, etc.) applied.
Hyperparameter tuning	Yes	`GridSearchCV` used for both `LogisticRegression` and `RandomForestClassifier` to select best hyperparameters.

RECIPE 2

1st Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No class distribution check
Sampling	Yes	80-20 split using `train_test_split()`
Outliers removal	Yes	Custom IQR-based function `remove_outlier()`
Check for duplicates	No	Not mentioned or applied
Imputation	Yes	Used `df.interpolate(method='linear')`
Drop columns	Yes	Dropped ID, SSN, etc.

Encoding	Yes	Used `LabelEncoder()` on all categorical columns
Create new columns	No (Modified existing)	Extracted years from `Credit_History_Age`
Feature selection	Yes	Based on correlation & VIF analysis
Scaling / Standardisation	Yes	Applied `RobustScaler` to `x_train`, `x_test`
Hyperparameter tuning	Partially	`C=100` manually selected; no GridSearchCV

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No label distribution analysis for `Credit_Score`
Sampling type	Random	`train_test_split(..., random_state=50)` used without stratification
Outliers removal	Yes	`remove_outlier()` removes rows based on 5th and 95th percentiles
Check for duplicates	No	No explicit check like `df.duplicated()`
Imputation of missing values	Use summary statistics (linear interpolation)	`df.interpolate(method='linear')` used
Drop columns	Yes	Columns like `ID`, `SSN`, `Name`, `Customer_ID`, `Month` were dropped
Encoding	Label Encoder	`LabelEncoder()` applied to categorical variables
Create new columns	No	All derived or transformed columns; none are standalone new columns

Feature selection	Yes	`Monthly_Inhand_Salary` and `Num_of_Delayed_Payment` removed after VIF
Data scaling	Yes	`RobustScaler()` used on train/test features
Hyperparameter tuning	No	Model trained with fixed `C=100`, no tuning/search procedure

3rd Prompt
Accuracy 11/11

Step	Used?	Details
Check for balanced data	Yes	Visual countplot of `Credit_Score`
Sampling type	Random	Used `train_test_split()` without stratification
Outlier removal	Yes	Via `remove_outlier()` function using 5th–95th percentile
Check for duplicates	No	No explicit check like `.duplicated()`
Missing value imputation	Yes	Linear interpolation (`interpolate()`)
Drop columns	Yes	Dropped ID, SSN, Customer_ID, Name, etc.
Encoding	LE	Used `LabelEncoder()` for all categorical features
Create new columns	No	No evidence of column creation
Feature selection	Yes	Removed features based on correlation and VIF
Standardization	Yes	`RobustScaler()` used on features
Hyperparameter tuning	No	No tuning, fixed `C=100` used for Logistic Regression

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details

Check for balanced data	Yes	`countplot` used for `Credit_Score` and other categorical variables
Sampling type	Random	`train_test_split(..., random_state=50)` without `stratify=`
Outliers removal	Yes	Custom `remove_outlier()` function applied using quantiles (5th–95th percentile)
Check for duplicates	No	No `duplicated()` or `drop_duplicates()` check observed
Imputation of missing values	use summary statistics	`interpolate(method='linear')` used for missing numeric values
Drop columns	Yes	Columns like `ID`, `Customer_ID`, `Name`, `Month`, SSN dropped without reuse
Encoding	Label Encoder	All categorical features encoded using `LabelEncoder()`
Create new columns	No	All modifications are derived (e.g., extracting year from `Credit_History_Age`)
Feature selection	Yes	Columns removed based on correlation/VIF (e.g., `Monthly_Inhand_Salary`, `Num_of_Delayed_Payment`)
Data scaling/standardisation	Yes	Applied `RobustScaler` to features before modeling
Hyperparameter tuning	No	Logistic Regression trained with fixed `C=100`, no GridSearch or RandomSearch used

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Distribution of `Credit_Score` (target) is explicitly visualized using countplots and discussed.
Sampling type	Random	`train_test_split()` used without `stratify`, so considered random sampling.

Outliers removal	Yes	Applied a quantile-based custom `remove_outlier` function filtering 5th–95th percentile of numeric columns.
Check for duplicates	No	No explicit check for duplicates like `.duplicated()` or `.drop_duplicates()` observed.
Imputation of missing values	mixture of imputation techniques	Categorical: replaced invalid strings and used interpolation. Numeric: used `interpolate(method='linear')`.
Drop columns	No	`'ID', 'Customer_ID', 'Month', 'Name', 'SSN'` dropped at the beginning
Encoding	Label Encoder	All categorical variables encoded with `LabelEncoder()`.
Create new columns	No	No new column derived from external data or unrelated transformations.
Feature selection	Yes	Removed features like `Monthly_Inhand_Salary`, `Num_of_Delayed_Payment` after VIF/correlation analysis.
Data scaling/standardisation	Yes	Applied `RobustScaler` to `x_train` and `x_test` before modeling.
Hyperparameter tuning	No	Logistic Regression applied with fixed parameter `C=100`, no `GridSearchCV` or equivalent used.

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Countplots for categorical columns include the target `Credit_Score`, showing its class distribution.
Sampling type	Random	`train_test_split(x, y, test_size=0.2, random_state=50)` without `stratify`.

Outliers removal	Yes	Quantile trimming via `remove_outlier`: keeps each numeric feature between its 5th and 95th percentiles.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` used.
Imputation of missing values	Use summary statistics	Numeric columns imputed with `df.interpolate(method='linear')`.
Drop columns	Yes	Dropped as irrelevant: `['ID','Customer_ID','Month','Name','SSN']` (not reused).
Encoding	Label Encoder	`LabelEncoder()` applied in-place to `Occupation`, `Type_of_Loan`, `Credit_Mix`, `Credit_History_Age`, `Payment_of_Min_Amount`, `Payment_Behaviour`, and `Credit_Score`.
Create new columns	No	Only transformations (e.g., extracting years from `Credit_History_Age`); no new features added.
Feature selection	Yes	Post-EDA/VIF: excluded highly collinear/less useful features (e.g., `Monthly_Inhand_Salary`, `Num_of_Delayed_Payment`); final model uses a selected subset (`mdf` list).
Data scaling/standardisation	Yes	`RobustScaler()` applied to `x_train` and `x_test`.
Hyperparameter tuning	No	`LogisticRegression(C=100)` with fixed parameters; no Grid/Random/Optuna search.

RECIPE 3

1st Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details

Check for balanced data	Yes	`pie_plot()` and countplot used to visualize class proportions
Sampling	Yes	`train_test_split` (80/20 split)
Outliers removal	Yes	IQR and explicit upper-limit filtering
Check for duplicates	Yes	Detected with `.duplicated().sum()`
Imputation	Yes	`fillna()` with median and random sampling
Drop columns	Yes	Dropped ID, Name, SSN, etc.
Encoding	Yes	Used both Label and One-Hot Encoding
Create new columns	No	Only cleaned existing ones
Feature selection	Yes	PCA (98% variance), VIF for multicollinearity
Scaling / Standardisation	Yes	Used `RobustScaler` for final input
Hyperparameter tuning	Partially	Manual tuning (e.g. `KNN(25)`, `DT(max_depth=3)`); no search CV used

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No label distribution check for `Credit_Score`
Sampling type	Random	`train_test_split(..., random_state=42)` used without stratification
Outliers removal	Yes	IQR method (Q1, Q3) and additional domain-based filtering for many numeric columns
Check for duplicates	Yes	`df.duplicated().sum()` used
Imputation of missing values	Mixture of imputation techniques	Used median, random sampling from valid categories, and `fillna()`

Drop columns	Yes	Dropped irrelevant columns such as SSN, Month, Name, Type_of_Loan, Credit_History_Age
Encoding	Mixture of encoding	Label Encoding (e.g., Credit_Score, Credit_Mix) + One-Hot Encoding (Occupation, Payment_Behaviour)
Create new columns	No	All column modifications were derived transformations only
Feature selection	Yes	PCA used to retain 98% variance + VIF used to drop collinear features
Data scaling	Yes	Used RobustScaler, StandardScaler, and MinMaxScaler; RobustScaler used for final model
Hyperparameter tuning	No	All models used fixed parameters (e.g., max_depth=3, K=25, C=100), no GridSearch or RandomSearch applied

3rd Prompt
Accuracy 11/11

Step	Used?	Details
Check for balanced data	Yes	Countplot and pie chart used to visualise label balance
Sampling type	Random	Used train_test_split(..., random_state=42)
Outlier removal	Yes	IQR filtering and hard-coded bounds used
Check for duplicates	Yes	Checked using df.duplicated().sum()
Missing value imputation	Yes	Used fillna(median), random.choice(...), and dropna()
Drop columns	Yes	Dropped irrelevant features like SSN, Name, Type_of_Loan, etc.
Encoding	LE, OHE	LabelEncoder style .replace(...) + pd.get_dummies() for categorical features
Create new columns	No	No new features added

Feature selection	Yes	Used `PCA(n_components=0.98)` to reduce feature space
Standardization	Yes	Used `RobustScaler` on numerical features
Hyperparameter tuning	No	All models used with fixed/default parameters

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`countplot` used for categorical variables including target `Credit_Score`
Sampling type	Random	`train_test_split(..., random_state=42)` used without `stratify=`
Outliers removal	Yes	IQR filtering applied + extreme value thresholds dropped manually (e.g., Age > 80, etc.)
Check for duplicates	Yes	`df.duplicated().sum()` used
Imputation of missing values	mixture of imputation techniques	Mode/random for categorical, median for numerical, and row-wise drop for >3 missing
Drop columns	Yes	Irrelevant fields like `ID`, `Name`, `SSN`, `Month`, etc. dropped
Encoding	mixture of encoding	Label Encoding + One-hot encoding via `pd.get_dummies()`
Create new columns	No	All transformations were replacements or derived fields (e.g., `.apply()` cleaning)
Feature selection	Yes	PCA applied to reduce dimensionality before modeling
Data scaling/standardisation	Yes	`RobustScaler` used, compared with `StandardScaler` and `MinMaxScaler`
Hyperparameter tuning	No	All models used default/fixed parameters; no GridSearchCV or RandomSearchCV applied

5th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Target label (`Credit_Score`) distribution visualized using pie plots; class balance mentioned and discussed in markdown.
Sampling type	Random	Used `train_test_split()` with fixed `random_state`, but no stratify, hence labeled as random sampling.
Outliers removal	Yes	Applied IQR-based filtering followed by hard thresholding on numeric columns (e.g., age, income, loans, cards).
Check for duplicates	Yes	Used `df.duplicated().sum()` to check and potentially remove duplicate records.
Imputation of missing values	mixture of imputation techniques	Used median for numerical columns, random sampling for categorical imputation, and row drop for heavy missingness.
Drop columns	No	Columns dropped (`Type_of_Loan`, `Credit_History_Age`, etc.) were removed after EDA, based on correlation/VIF, so considered feature selection.
Encoding	mixture of encoding	Label Encoding for ordinal (`Credit_Score`, `Credit_Mix`) and One-hot encoding for nominal (`Occupation`, `Payment_Behaviour`).
Create new columns	No	No new columns added beyond scaling/encoding; transformations only.
Feature selection	Yes	Dropped columns based on VIF, PCA used to reduce dimensionality to 98% explained variance.
Data scaling/standardisation	Yes	Applied `RobustScaler` to numerical features and visualized impact across `StandardScaler`, `MinMaxScaler`, and `RobustScaler`.

Hyperparameter tuning	No	No GridSearchCV or similar used; all models (KNN, LR, NB, DT, RF, Neural Net) trained with default or manually set parameters.

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Pie chart of `Credit_Score` via `pie_plot(..., ['Credit_Mix','Payment_of_Min_Amount','Payment_Behaviour','Credit_Score'], ...)`.
Sampling type	Random	`train_test_split(x_reduced, y_clean, test_size=0.2, random_state=42)` with no `stratify`.
Outliers removal	Yes	Percentile trimming per numeric column (keep 0.05–99.95th) in a loop, then hard caps (e.g., drop `Age ≥ 80`, `Annual_Income ≥ 500000`, etc.).
Check for duplicates	Yes	Checked with `df_c.duplicated().sum()`.
Imputation of missing values	Mixture of imputation techniques	Categorical filled by random sampling of valid categories; numeric filled with medians (e.g., `Monthly_Inhand_Salary`, `Num_of_Delayed_Payment`, etc.) and later forward fill; also dropped rows with ≥3 missing values.
Drop columns	Yes	Dropped as irrelevant: `['ID','Customer_ID','Month','Name','Type_of_Loan','Credit_History_Age','SSN']` (not reused).
Encoding	Mixture	Label-style mappings for `Credit_Score`, `Credit_Mix`, `Payment_of_Min_Amount`; one-hot via `pd.get_dummies` for `Occupation`, `Payment_Behaviour`.
Create new columns	No	One-hot dummies are derived from existing columns;not counted as “new”.

Feature selection	Yes	Dimensionality reduction with `PCA(n_components=0.98)` (reduces feature space before modelling).
Data scaling/standardisation	Yes	`RobustScaler()` applied to numeric features (final dataset built with robust-scaled numerics).
Hyperparameter tuning	No	Models trained with fixed settings (e.g., `KNeighborsClassifier(25)`, simple `LogisticRegression`, NN with fixed architecture); no Grid/Random search.

RECIPE 4

1st Prompt
Accuracy 8/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No pie chart or class plot; evaluated post-model
Sampling	Yes	Random split using `train_test_split`
Outliers removal	Yes	Visualized (kde, box), capped, and filtered
Check for duplicates	No	Not addressed in this notebook
Imputation	Yes	Used custom functions and mode by Customer_ID
Drop columns	Yes	Removed IDs, names, and irrelevant fields
Encoding	Yes	`LabelEncoder` for categorical fields, month mapped manually
Create new columns	Yes	Loan types exploded into binary columns
Feature selection	Yes	PCA with 98% explained variance
Scaling / Standardisation	Yes	`MinMaxScaler` used globally
Hyperparameter tuning	Yes	Manually defined per model, no search CV

2nd Prompt

Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No label distribution check for `Credit_Score`
Sampling type	Random	`train_test_split` used with fixed `random_state` and no stratification
Outliers removal	Yes	Applied hard-coded threshold filtering on numeric columns (e.g., Age > 60, Income > 165000)
Check for duplicates	No	No `duplicated()` or equivalent check was present
Imputation of missing values	Mixture of imputation techniques	Imputation using mode per customer ID, column-wise mode/median, and fixed values like `"Not Specified"`
Drop columns	Yes	Dropped `ID`, `Customer_ID`, `Name`, `SSN`, and `Type_of_Loan` after encoding
Encoding	Label Encoding + Multi-hot	Label encoding for multiple categorical features; multi-hot encoding (manually) for `Type_of_Loan`
Create new columns	Yes	One-hot style binary columns created for each loan type from `Type_of_Loan`
Feature selection	Yes	PCA used to retain 98% variance
Data scaling	Yes	Used `MinMaxScaler` on all numeric features
Hyperparameter tuning	No	Fixed hyperparameters used; no `GridSearchCV` or `RandomizedSearchCV`

3rd Prompt
Accuracy 7/11

Step	Used?	Details
Check for balanced data	No	No visualization or check present
Sampling type	Random	`train_test_split(..., random_state=1234)`

Outlier removal	Yes	Numerous domain-based thresholds and filters
Check for duplicates	No	Not checked
Missing value imputation	Yes	Mode-based conditional imputation
Drop columns	Yes	Dropped unnecessary or PII columns
Encoding	LE	LabelEncoder used on multiple categorical variables
Create new columns	Yes	`Type_of_Loan` split into multiple binary columns
Feature selection	Yes	PCA with `n_components=0.98`
Standardization	Yes	`MinMaxScaler()` applied
Hyperparameter tuning	No	Models used with fixed parameters

4th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No visualisation or count analysis shown for class balance (e.g., value_counts or countplot)
Sampling type	Random	`train_test_split(...,` `random_state=1234)` used repeatedly without `stratify=`
Outliers removal	Yes	Extensive outlier filtering applied using thresholds across multiple columns
Check for duplicates	No	No use of `.duplicated()` or `.drop_duplicates()`
Imputation of missing values	mixture of imputation techniques	Used mode imputation by customer ID, manual filling, median, and random mode for objects
Drop columns	Yes	Columns like `ID`, `Name`, `Customer_ID`, and `SSN` dropped
Encoding	mixture of encoding	Used Label Encoding for multiple categorical columns + multi-hot encoding for loan types

Create new columns	No	Multi-label fields like loan types decomposed, but still derived from existing info
Feature selection	Yes	PCA applied to reduce dimensionality before modeling
Data scaling/standardisation	Yes	Applied `MinMaxScaler` before PCA and modeling
Hyperparameter tuning	No	All models use fixed parameters; no GridSearchCV or RandomSearchCV applied

5th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No explicit check using `value_counts()` or class-distribution plots for the target variable `Credit_Score`.
Sampling type	Random	Used `train_test_split()` with fixed `random_state=1234`, no stratification used.
Outliers removal	Yes	Applied hard thresholds (e.g., `Age > 60`, `EMI < 5000`) and custom filters using `.apply()` and `.drop()` based on visual inspection and KDE plots.
Check for duplicates	No	No `.duplicated()` or `.drop_duplicates()` check observed.
Imputation of missing values	mixture of imputation techniques	Used mode imputation per `Customer_ID`, column-wise median, and hardcoded fallbacks; multiple strategies applied depending on column type.
Drop columns	Yes	`'ID'`,`'Customer_ID'`,`'Name'`,`'SSN'` columns dropped.
Encoding	mixture of encoding	Label Encoding (`LabelEncoder`) for ordered categories; one-hot encoding for multi-loan types via custom parsing.

Create new columns	Yes	Created individual binary columns for loan types parsed from multi-valued `Type_of_Loan`.
Feature selection	No	No columns dropped based on feature importances or correlation.
Data scaling/standardisation	Yes	Applied `MinMaxScaler` before PCA and model training.
Hyperparameter tuning	No	All models used default or manually set parameters (e.g., `max_depth=12`); no `GridSearchCV`, `RandomizedSearchCV`, or tuning loop observed.

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No value counts or class-distribution plot for `Credit_Score`.
Sampling type	Random	`train_test_split(..., test_size=0.3, random_state=1234)` used repeatedly without `stratify`.
Outliers removal	Yes	Rule-based filtering per feature (e.g., `Monthly_Inhand_Salary < 13500`, `Num_of_Delayed_Payment < 150 & ≥ 0`, caps on `Num_Credit_Inquiries`, `Age` outside bounds dropped, etc.).
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` calls.
Imputation of missing values	Mixture of imputation techniques	Per-customer mode imputation via custom functions; global mode for categoricals; text placeholder `'Not Specified'` for `Type_of_Loan`.
Drop columns	Yes	Dropped `SSN` early; later dropped `Type_of_Loan` after expanding indicators; finally removed `['ID','Customer_ID','Name']`.

Encoding	Mixture	Manual one-hot style expansion for `Type_of_Loan` + `LabelEncoder` for `Occupation`, `Credit_Mix`, `Payment_Behaviour`, `Payment_of_Min_Amount`; month name → integer map.
Create new columns	No	Loan indicator columns are derived from `Type_of_Loan`, so not counted as "new".
Feature selection	No	PCA (`n_components=0.98`) computed but not used for modelling; no correlation/model-based drops tied to modelling.
Data scaling/standardisation	Yes	`MinMaxScaler()` fitted on features before modelling.
Hyperparameter tuning	No	Fixed estimator settings; no `GridSearchCV`/`RandomizedSearchCV` used.

RECIPE 5

1st Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used pie chart (`plt.pie`) and `sns.countplot()` to visually inspect `Credit_Score` class distribution.
Sampling type	Random	Used `train_test_split()` without `stratify`, so default random sampling.
Outliers removal	No	Although `sns.boxplot()` was used for visual inspection, no rows were removed or clipped.
Check for duplicates	No	No code to check duplicates using `duplicated()` or `drop_duplicates()`.
Imputation of missing values	Mixture of imputation techniques	Used: `ffill` (forward fill) for text fields`mean` for some numeric fields`dropna(thresh=23)` to drop rows with too many missing values

Drop columns	Yes	Dropped `"ID"` column explicitly via `del train["ID"]`.
Encoding	Label Encoding	Applied `LabelEncoder` to all `object`-type columns using a `for` loop.
Create new columns	No	No new columns were created in the code.
Feature selection	No	All features used post-cleaning. No dimensionality reduction or feature importance selection performed.
Data scaling / standardisation	Yes	Applied `RobustScaler` using `fit_transform()` on both train and test datasets.
Hyperparameter tuning	No	Parameters like `C=100`, `n_neighbors=13`, `max_depth=4`, and `max_features=5` were hardcoded without tuning loops or grid search.

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No check for class distribution before modeling
Sampling type	Yes	`train_test_split` used for random sampling (no stratification or resampling)
Outliers removal	No	Outliers were visualized with boxplots, but not removed in code
Check for duplicates	No	No duplicate check or removal in the code
Imputation of missing values	Yes	Mixture of imputation techniques: forward fill (`ffill`), mean, and hardcoded values (`0`)
Drop columns	Yes	`ID` column was dropped completely
Encoding	Label Encoder	All object columns encoded using `LabelEncoder`
Create new columns	No	No new columns were added beyond existing dataset (Prompt-3 rule)

Feature selection	No	All columns (except dropped ones) were kept; no correlation/VIF-based filtering
Data scaling/standardisation	Yes	Used `RobustScaler` for both training and test sets
Hyperparameter tuning	No	Model hyperparameters like `C`, `n_neighbors`, `max_depth`, etc., were manually set, not optimized systematically

3rd Prompt
Accuracy 11/11

Step	Used?	Details
Check for balanced data	Yes	`countplot` and `pie chart` used for class balance
Sampling type	Random	`train_test_split(..., test_size=0.3)`
Outlier removal	No	No IQR or quantile filtering
Check for duplicates	No	Not performed
Missing value imputation	Yes	Used `dropna(thresh=23)`, `.fillna()`, `ffill()`
Drop columns	Yes	`ID` column dropped
Encoding	LE	Used `LabelEncoder` on all object columns
Create new columns	No	No feature engineering
Feature selection	No	No selection or dimensionality reduction
Standardization	Yes	`RobustScaler()` used
Hyperparameter tuning	No	All models used with fixed hyperparameters

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`value_counts()` and `countplot()` on `Credit_Score` used for balance analysis

Sampling type	Random	`train_test_split(..., test_size=0.3)` without `stratify=`
Outliers removal	No	No statistical or visual filtering used to drop or clip outliers
Check for duplicates	No	No `duplicated()` or `drop_duplicates()` found
Imputation of missing values	mixture of imputation techniques	Used `dropna(thresh=23)`, `.fillna()` with `mean`, `ffill`, and hard-coded values
Drop columns	Yes	`ID` column explicitly dropped
Encoding	Label Encoder	`LabelEncoder` applied across all object columns
Create new columns	No	No new feature creation detected
Feature selection	No	All features used directly; no dimensionality reduction or correlation-based pruning
Data scaling/standardisation	Yes	`RobustScaler` applied to both train and test features
Hyperparameter tuning	No	All models used fixed parameters without tuning (e.g., fixed depth, k, C)

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Target variable `Credit_Score` distribution visualized via `value_counts()` and pie chart/countplot.
Sampling type	Random	`train_test_split()` used without `stratify`, indicating random sampling.
Outliers removal	No	Boxplots plotted, but no filtering or dropping of outliers was implemented.
Check for duplicates	No	No `.duplicated()` or equivalent check observed.

Imputation of missing values	mixture of imputation techniques	Used `.mean()` (numerical), `.ffill()` (categorical), and `.dropna(thresh=23)` to selectively retain rows.
Drop columns	Yes	Only `ID` was explicitly dropped; its information was not reused or transformed.
Encoding	Label Encoder	All object-type categorical columns encoded using `LabelEncoder()`.
Create new columns	No	No new columns added during processing.
Feature selection	No	No evidence of dropping columns based on correlation, importance, or post-EDA analysis.
Data scaling/standardisation	Yes	Applied `RobustScaler` to train and test features before modeling.
Hyperparameter tuning	No	All models used fixed parameters (e.g., `max_depth=4`, `C=100`, `n_neighbors=13`); no tuning loop or search applied.

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Pie chart and countplot of `Credit_Score` (`plt.pie`, `sns.countplot`).
Sampling type	Random	`train_test_split(train, test_size=0.3)` without `stratify`.
Outliers removal	No	Only boxplots/EDA; no IQR/quantile/rule-based filtering.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` used.
Imputation of missing values	Mixture of imputation techniques	Dropped rows with `dropna(thresh=23)`; replaced special string in `Monthly_Balance`; used forward fill for several columns and mean for others (e.g., `Monthly_Inhand_Salary`, `Num_Credit_Inquiries`).

Drop columns	Yes	Dropped `ID` (not reused).
Encoding	Label Encoder	Loop applies `LabelEncoder()` to all object columns (in-place), including `Credit_Score`.
Create new columns	No	No genuinely new features; only in-place recoding.
Feature selection	No	No correlation/model-based or post-EDA drops for modelling.
Data scaling/standardisation	Yes	`RobustScaler()` applied to `train_x` and `test_x`.
Hyperparameter tuning	No	Fixed model params (e.g., `LogisticRegression(C=100)`, `KNN(n_neighbors=13)`); no Grid/Random/Optuna search.

RECIPE 6

1st Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No `value_counts()` or class balance checks were done on `Credit_Score`
Sampling type	Stratified	`train_test_split()` was used without `stratify=y`, so this is random sampling (`Instruction c`)
Outliers removal	Yes	Multiple logical filters applied on numerical columns like `Age`, `Annual_Income`, etc.
Check for duplicates	No	No `.duplicated()` or `.drop_duplicates()` used
Imputation of missing values	Mixture of techniques	- `dropna(thresh=26)` to drop rows - Remaining missing values imputed via `np.random.choice()` from non-null list for each column

Drop columns	Yes	Dropped columns not needed using subset selection, e.g. `data = data[['Month', ..., 'Credit_Score']]`
Encoding	Label Encoding	Used `LabelEncoder()` on all object-type columns including target
Create new columns	No	No new columns were created
Feature selection	Yes	Based on correlation with `Credit_Score`, selected top variables into `mdf` dataframe
Data scaling / standardisation	Yes	Used `RobustScaler()` to scale features before modeling
Hyperparameter tuning	No	Model parameters were hardcoded; no `GridSearchCV` or `RandomizedSearchCV` used

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No code checks class distribution of `Credit_Score`
Sampling type	Yes (random)	`train_test_split` without stratification
Outliers removal	Yes	Applied threshold filters on multiple columns (e.g., `Age`, `Income`)
Check for duplicates	No	No use of `.duplicated()` or `.drop_duplicates()`
Imputation of missing values	Yes (mixture)	Random choice imputation + row drop based on `thresh`
Drop columns	Yes	Columns dropped permanently (e.g., `ID`, `SSN`, etc.)
Encoding	Label Encoding	`LabelEncoder` used on all object columns
Create new columns	No	No columns created; all transformations stayed in existing columns
Feature selection	Yes	Feature selection done via correlation with `Credit_Score`

Data scaling or standardisation	Yes (RobustScaler)	Applied `RobustScaler` to both `x_train` and `x_test`
Hyperparameter tuning	No	Hyperparameters were manually specified, not tuned via grid/random search

3rd Prompt
Accuracy 11/11

Step	Used?	Details
Check for balanced data	Yes	Countplot by `Credit_Score`
Sampling type	Random	`train_test_split` with test size 0.25
Outlier removal	Yes	Conditional filtering based on domain knowledge
Check for duplicates	No	Not performed
Missing value imputation	Mixture	`dropna(thresh=26)` + `fillna(random.choice(...))`
Drop columns	Yes	Dropped unused fields
Encoding	LE	`LabelEncoder()` on all categorical columns
Create new columns	No	No derived variables
Feature selection	Yes	Based on correlation with target
Standardization	Yes	`RobustScaler()` used
Hyperparameter tuning	No	Manual setting of model parameters

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`countplot` and `displot` of `Credit_Score` shown
Sampling type	Random	`train_test_split(..., random_state=42)` without `stratify=`

Outliers removal	Yes	Numerous filters applied to cap or remove extreme values (e.g., Age ≤ 100, EMI < 75,000)
Check for duplicates	No	No use of `duplicated()` or `drop_duplicates()`
Imputation of missing values	mixture of imputation techniques	Used row drops (`thresh=26`), random fill from valid values, and manual mode/mean filling
Drop columns	Yes	Dropped ID, Customer_ID, and other metadata not used in modeling
Encoding	Label Encoder	Applied LabelEncoder to all object columns including target
Create new columns	No	All transformations (e.g., `.str.replace`, `.fillna`, label encoders) were within existing columns
Feature selection	Yes	Features selected based on correlation with `Credit_Score` and plotted as bar chart
Data scaling/standardisation	Yes	Used `RobustScaler` on training and test data
Hyperparameter tuning	No	All classifiers used fixed parameters; no use of GridSearch or RandomSearchCV

5th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used `sns.countplot()` and distribution plots on `Credit_Score`; classes are visualized and explicitly analysed.
Sampling type	Random	Used `train_test_split()` with `random_state=42`, no stratification.
Outliers removal	Yes	Applied hard threshold filtering for most numeric features (e.g., `Age <= 100`, `Income <= 300000`, `Credit Limit <= 30`).

Check for duplicates	No	No `duplicated()` check or removal logic seen.
Imputation of missing values	mixture of imputation techniques	Dropped rows with extreme missingness; remaining NaNs filled via `np.random.choice` from non-missing values (for both categorical and numeric).
Drop columns	No	Columns dropped (`ID`, `SSN`, etc.) not shown in the cleaned version explicitly; dropped only internally after correlation/EDA.
Encoding	Label Encoder	Used `LabelEncoder()` for all categorical columns.
Create new columns	No	No creation of new columns from existing features (e.g., parsing or engineered variables).
Feature selection	Yes	Used correlation analysis to select top 14 features most associated with `Credit_Score`; selected manually into final model input.
Data scaling/standardisation	Yes	Applied `RobustScaler()` before model training.
Hyperparameter tuning	No	Models trained using manually chosen hyperparameters; no automated tuning (GridSearchCV, Optuna, etc.).

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Multiple EDA plots with `hue='Credit_Score'` (displots/countplots) allow visual inspection of class balance.
Sampling type	Random	`train_test_split(x, y, test_size=0.25, random_state=42)` with no `stratify`.

Outliers removal	Yes	Extensive rule-based filters per feature (e.g., `0 ≤ Age ≤ 100`, `Annual_Income ≤ 300000`, `Num_Credit_Card ≤ 1000`, …).
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` used.
Imputation of missing values	Mixture of imputation techniques	Row-wise thresholds `dropna(thresh=26)` then `thresh=24`, plus random-sample fills for many columns (e.g., `Monthly_Inhand_Salary`, `Type_of_Loan`, `Num_Credit_Inquiries`, …).
Drop columns	Yes	Restricted feature set removes identifiers (`ID`, `Customer_ID`, `Name`, SSN) from analysis.
Encoding	Label Encoder	`LabelEncoder()` applied in-place to object columns and target (`Credit_Score`).
Create new columns	No	No genuinely new features (encodings are derived from existing columns).
Feature selection	Yes	Correlation-driven selection into `mdf` after heatmap/importance inspection.
Data scaling/standardisation	Yes	`RobustScaler()` fitted on train and (incorrectly) re-fitted on test, but scaling is present.
Hyperparameter tuning	No	Fixed hyperparameters across models; no Grid/Random/Optuna search.

RECIPE 7
1st Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used bar plots and factor plots for `Credit_Score` distributions in relation to categorical features.
Sampling type	Random	`train_test_split()` used without `stratify` — data split with shuffle=True.

Outliers removal	Yes	Used Tukey method (`detect_outliers()`) to find and drop multiple outliers based on IQR.
Check for duplicates	No	No `duplicated()` or `drop_duplicates()` was used.
Imputation of missing values	Mixture of techniques	Used summary statistics (mean/median), conditional imputation (by `Credit_Score`), and dropped some columns.
Drop columns	Yes	Dropped 13 columns including ID, Customer_ID, SSN, Name, etc. using `dataset.drop(...)`.
Encoding	Mixture of encoding	Used both one-hot encoding (`pd.get_dummies`) and manual numeric mapping for target variable.
Create new columns	Yes	Created 8 new binary columns from multi-label `Type_of_Loan` field (e.g., `Auto_Loan`, `Personal_Loan`).
Feature selection	Yes	Removed features with poor predictive value and engineered new relevant ones.
Data scaling or standardization	Yes (via log transformation)	Applied log transforms on many skewed columns (e.g., Age, Salary, EMI) to reduce skewness — a form of normalization.
Hyperparameter tuning	No	Parameters (e.g., `n_neighbors`, `n_estimators`) were manually set; no search or optimization loop (`GridSearchCV`, etc.) used.

2nd Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No check for `Credit_Score` class balance
Sampling type	Random	`train_test_split(shuffle=True)`
Outliers removal	Yes	Tukey's method (`IQR`) used on numeric features
Check for duplicates	No	No duplicate check performed

Imputation of missing values	Mixture of techniques	Used class-wise mean, median, and drop rows
Drop columns	Yes	Dropped columns like `ID`, `Name`, `SSN`, etc.
Encoding	Mixture of encoding	One-hot encoding + manual label conversion for target
Create new columns	Yes	Multi-hot encoding derived from `Type_of_Loan`
Feature selection	Yes	Columns manually selected based on domain/EDA
Data scaling	No	Only log transformation applied; no scaler used
Hyperparameter tuning	No	Models used default or manually set hyperparameters

3rd Prompt
Accuracy 10/11

Step	Used?	Details
Check for balanced data	Yes	Countplots by `Credit_Score` used
Sampling type	Random	`train_test_split(..., shuffle=True)`
Outlier removal	Yes	IQR-based filtering with `detect_outliers()`
Check for duplicates	No	Not performed
Missing value imputation	Yes	Median, mean, and logic-based fills
Drop columns	Yes	Many irrelevant columns dropped
Encoding	OHE, Manual	One-hot encoding and label encoding for target
Create new columns	Yes	Binary columns from `Type_of_Loan`
Feature selection	Yes	Correlation and manual pruning
Standardization	No	No scaling applied
Hyperparameter tuning	No	Models used with fixed parameters

4th Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Distribution of target variable `Credit_Score` checked using `factorplot` and count plots
Sampling type	Random	`train_test_split(..., shuffle=True)` used without stratify
Outliers removal	Yes	Tukey method via IQR used with `detect_outliers()` and dropped using index list
Check for duplicates	No	No `.duplicated()` or `.drop_duplicates()` used
Imputation of missing values	use summary statistics	Mean by class, medians used for imputing missing values in multiple columns
Drop columns	Yes	Columns like `ID`, `Customer_ID`, `SSN`, etc. dropped without being used elsewhere
Encoding	mixture of encoding	`pd.get_dummies` for some features; manual label encoding for target `Credit_Score`
Create new columns	Yes	New binary columns created from `Type_of_Loan` decomposition
Feature selection	No	No explicit selection based on correlation, importance, or post-EDA filtering
Data scaling/standardisation	No	Only log transformations applied; no scaler used
Hyperparameter tuning	No	Multiple models compared, but no GridSearchCV/RandomSearchCV/Optuna used

5th Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details

Check for balanced data	Yes	`value_counts()` and `factorplot()` used on `Credit_Score`
Sampling type	Random	`train_test_split` used without stratification
Outliers removal	Yes	Tukey method applied via `detect_outliers()` on numerical features
Check for duplicates	No	No code to check duplicates
Imputation of missing values	mixture of imputation techniques	Used group-wise mean, median, and column-wise fillna for multiple columns
Drop columns	No	Columns dropped post EDA based on domain/model utility
Encoding	One hot encoding	Used `pd.get_dummies()` and manual encoding for target variable
Create new columns	No	Created new binary columns for each loan type in `Type_of_Loan` not really creating new information.
Feature selection	Yes	Columns removed post-EDA and correlation analysis
Data scaling/standardisation	No	Applied log transformation to reduce skewness (not scaling)
Hyperparameter tuning	No	All models use fixed parameters without GridSearch or similar tuning

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Multiple count/facet plots using `Credit_Score` (e.g., `sns.factorplot('Credit_Score', col=..., kind='count')`) and KDEs by `Credit_Score` to inspect target distribution.

Sampling type	Random	`train_test_split(X, Y, test_size=0.3, random_state=27, shuffle=True)` without `stratify`.
Outliers removal	Yes	Tukey IQR method via `detect_outliers(...)` over all numeric columns; identified indices dropped from the dataset.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` usage found.
Imputation of missing values	Use summary statistics (mean/median)	`Monthly_Inhand_Salary` imputed by mean per Credit_Score; `Num_of_Delayed_Payment`, `Amount_invested_monthly`, `Num_Credit_Inquiries` imputed with median.
Drop columns	Yes	Dropped identifiers/others without replacement: `['ID','Customer_ID','Name','SSN','Num_of_Loan','Credit_Utilization_Ratio','Credit_History_Age','Payment_Behaviour','Annual_Income','Monthly_Balance','Num_Bank_Accounts','Num_Credit_Card','Credit_Mix']`. (`Type_of_Loan` dropped after deriving indicators; not counted here.)
Encoding	Mixture	`pd.get_dummies` on `Month`, `Occupation`, `Payment_of_Min_Amount`; target `Credit_Score` mapped to numeric `Target` (0/1/2).
Create new columns	No	Loan indicator columns (`Auto_Loan`, `...`, `Payday_Loan`) are derived from `Type_of_Loan`, so not counted as “new”.
Feature selection	Yes	Columns removed after EDA within preprocessing (post-visualisation drop counts as feature selection).
Data scaling/standardisation	No	Only in-place log transforms of several numerics (log ≠ scaling); no `StandardScaler/MinMaxScaler`.
Hyperparameter tuning	No	Tried fixed settings across models (LR, KNN k=1/3/5/7, DT, RF with set `n_estimators`), no Grid/Random/Optuna search.

RECIPE 8

1st Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used `value_counts()` and `countplot()` on `Credit_Score` to visualize class balance.
Sampling type	Random	`train_test_split()` used with `shuffle=True` and no stratification.
Outliers removal	No	No use of filters, IQR, Z-score, or outlier removal techniques seen.
Check for duplicates	No	No `.duplicated()` or `.drop_duplicates()` call in the code.
Imputation of missing values	Use summary statistics	Used `SimpleImputer(strategy='mean')` to fill in missing numerical values.
Drop columns	Yes	Dropped multiple columns explicitly (e.g., `Type_of_Loan`, `ID`, `Customer_ID`, etc.)
Encoding	Mixture of encoding	Used `LabelEncoder` for target variable, `LeaveOneOutEncoder` for categorical variables, and mean encoding for numericals.
Create new columns	No	No new features were created from scratch or transformed into binary/multi-class representations.
Feature selection	Yes	Removed irrelevant columns after correlation analysis; selected only numerically relevant ones.
Data scaling or standardisation	No	No scaler used (e.g., RobustScaler, StandardScaler, or MinMaxScaler).
Hyperparameter tuning	No	Model parameters were default; no `GridSearchCV`, `RandomSearchCV`, or manual tuning attempts.

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	No	No class distribution check for `Credit_Score`

Sampling type	Random	`train_test_split(..., shuffle=True)`
Outliers removal	No	Visualized via boxplots, but no removal applied
Check for duplicates	No	No `.duplicated()` or `.drop_duplicates()` used
Imputation of missing values	Use summary statistics	Applied `SimpleImputer(strategy='mean')`
Drop columns	Yes	Dropped `ID`, `Customer_ID`, `Month`, `SSN`, `Amount_invested_monthly`, `Type_of_Loan`, `Name`, etc.
Encoding	Mixture of encoding	Label Encoding, Leave-One-Out, and Mean encoding used
Create new columns	No	Only transformed existing columns; no new ones added
Feature selection	Yes	Used correlation matrix + dropped low relevance features
Data scaling/standardisation	No	No scaler (e.g., MinMax, RobustScaler) used
Hyperparameter tuning	No	All model parameters used default or manually set

3rd Prompt
Accuracy 11/11

Step	Used?	Details
Check for balanced data	Yes	`countplot`, `value_counts()`
Sampling type	Random	`train_test_split(..., shuffle=True)`
Outlier removal	No	Outliers plotted, but not removed
Check for duplicates	No	Not done
Missing value imputation	Yes	`SimpleImputer(strategy='mean')`
Drop columns	Yes	Many irrelevant columns removed

Encoding	LE, LOO, Mean	Mixed encodings: label, leave-one-out, and mean
Create new columns	No	No column creation
Feature selection	Yes	Columns dropped manually
Standardization	No	No scaler used
Hyperparameter tuning	No	No tuning or cross-validation

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`value_counts()` and `countplot()` used for target `Credit_Score`
Sampling type	Random	`train_test_split(..., shuffle=True, random_state=42)` without `stratify=`
Outliers removal	No	Visualizations used (boxplots), but no filtering or dropping based on thresholds or IQR
Check for duplicates	No	No `.duplicated()` or `.drop_duplicates()` used
Imputation of missing values	use summary statistics	`SimpleImputer(strategy='mean')` applied to all numeric columns
Drop columns	Yes	Dropped multiple columns: `ID`, `Customer_ID`, `Month`, SSN, `Amount_invested_monthly`, etc.
Encoding	mixture of encoding	LabelEncoder on target + LeaveOneOutEncoder and mean encoding on categorical/numeric features
Create new columns	No	No column was newly created; all were encoded or derived by transformation
Feature selection	Yes	Multiple columns dropped post-EDA manually (e.g., `Name`, `Num_of_Loan`, `Outstanding_Debt`)

Data scaling/standardisation	No	No scaler (e.g., StandardScaler, MinMaxScaler, RobustScaler) used
Hyperparameter tuning	No	All classifiers used with default parameters; no tuning libraries used

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used `value_counts()` and `countplot()` to inspect class distribution of `Credit_Score`.
Sampling type	Random	Used `train_test_split()` with `shuffle=True`, no stratification.
Outliers removal	No	Visualized outliers via `boxplot`, but no filtering or dropping implemented.
Check for duplicates	No	No `.duplicated()` check or removal logic observed.
Imputation of missing values	use summary statistics	Applied `SimpleImputer(strategy='mean')` on numeric features.
Drop columns	Yes	Dropped columns like `Type_of_Loan`, `ID`, `SSN`, `Customer_ID`, `Month`, etc., directly using `drop()`.
Encoding	mixture of encoding	Used `LabelEncoder` for target and `LeaveOneOutEncoder` & `mean encoding` for other categorical/numeric features.
Create new columns	No	No new columns introduced; encoding only replaced existing columns.
Feature selection	Yes	Columns dropped based on correlation and domain knowledge.
Data scaling/standardisation	No	No standardisation or scaling applied.

Hyperparameter tuning	No	All models (LogReg, Ridge, KNN, Tree, SVC) used default parameters; no search or tuning attempted.

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`train['Credit_Score'].value_counts()` and `sns.countplot(x='Credit_Score')` used to inspect target distribution.
Sampling type	Random	`train_test_split(X, y, test_size=0.33, shuffle=True, random_state=42)` with no `stratify`.
Outliers removal	No	Only boxplots/EDA; no IQR/quantile/rule-based filtering performed.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` present.
Imputation of missing values	Use summary statistics (mean)	`SimpleImputer(strategy='mean')` applied to features for train/test.
Drop columns	Yes	Dropped `Type_of_Loan`, then `['ID','Customer_ID','Month','SSN','Amount_invested_monthly']`, and later `['Name','Annual_Income','Num_of_Loan','Outstanding_Debt']` (not reused).
Encoding	Mixture	Target `Credit_Score` label-encoded; categorical features encoded via `category_encoders.LeaveOneOutEncoder`; additional custom mean/target encoding for several numeric-like columns.
Create new columns	No	Encodings overwrite in place; no one-hot expansion or genuinely new features added.
Feature selection	Yes	Columns removed are identifiers or manual exclusions

Data scaling/standardisation	No	No scaler (`StandardScaler`, `MinMaxScaler`, etc.) used.
Hyperparameter tuning	No	Models trained with fixed/default params; no Grid/Random/Optuna search.

RECIPE 9

1st Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used `value_counts()`, calculated normalized class distribution, applied class weights to handle imbalance.
Sampling type	Random	`train_test_split(test_size=0.2, random_state=42)` without stratification.
Outliers removal	No	No filtering or removal of outliers observed (e.g., no z-score, IQR, or logical constraints).
Check for duplicates	No	No use of `.duplicated()` or `.drop_duplicates()` seen in the code.
Imputation of missing values	No	No explicit imputation (e.g., SimpleImputer, fillna, etc.) observed — data is already pre-cleaned.
Drop columns	Yes	Dropped `['ID', 'Customer_ID', 'Month', 'Name', 'SSN']` and unused features during feature selection.
Encoding	Mixture of encoding	Used `LabelEncoder`, `OrdinalEncoder`, `OneHotEncoder`, and a custom `GetDummies` transformer.
Create new columns	Yes	Used custom `GetDummies` to decompose multi-label text fields into dummy binary features.
Feature selection	Yes	Used XGBoost feature importance, permutation importance, and Yellowbrick to select top features.

Data scaling or standardisation	Yes	Used `MinMaxScaler` to scale data for models like Logistic Regression.
Hyperparameter tuning	Yes	Used `GridSearchCV` with 5-fold cross-validation for all models (Logistic Regression, Random Forest, XGBoost).

2nd Prompt
Accuracy 10/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Class imbalance explicitly handled with class weights and sample weights
Sampling type	Random	`train_test_split(random_state=42)` used
Outliers removal	No	Outliers were visualized but not removed
Check for duplicates	No	No code for checking or dropping duplicates
Imputation of missing values	Ignore	Dataset was pre-cleaned; no missing value handling seen
Drop columns	Yes	Dropped `ID`, `Customer_ID`, `Month`, `Name`, `SSN` and others
Encoding	Mixture of encoding	Used LabelEncoder, OrdinalEncoder, OneHotEncoder, custom GetDummies
Create new columns	Yes	Dummy variables created via multi-hot GetDummies transformer
Feature selection	Yes	Based on XGBoost feature importance + permutation importance
Data scaling/standardisation	Yes	MinMaxScaler used for numerical features
Hyperparameter tuning	Yes	GridSearchCV for Logistic Regression, Random Forest, XGBoost

3rd Prompt
Accuracy 10/11

Step	Used?	Details

Check for balanced data	Yes	`value_counts`, visualizations
Sampling type	Random	`train_test_split(..., random_state=...)`
Outlier removal	No	Visualized only
Check for duplicates	No	Not checked
Missing value imputation	No	Not performed in this cleaned dataset
Drop columns	Yes	Dropped many ID/irrelevant fields
Encoding	LE, OHE, Ordinal, Dummy	All types implemented
Create new columns	Yes	Dummy variables extracted from text
Feature selection	Yes	Via XGBoost importance and pruning
Standardization	Yes	`MinMaxScaler()` used
Hyperparameter tuning	Yes	`GridSearchCV` used for 3 models

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`value_counts(normalize=True)` and class distribution plots used for `Credit_Score`
Sampling type	Random	`train_test_split(X, y, test_size=0.2, random_state=42)` with no `stratify`
Outliers removal	No	No filtering, capping, or IQR-based removal observed
Check for duplicates	No	No `duplicated()` or `drop_duplicates()` used
Imputation of missing values	none	No missing value imputation applied; cleaned dataset is used as stated

Drop columns	Yes	Columns like `ID`, `Customer_ID`, `Month`, `Name`, `SSN` were dropped
Encoding	mixture of encoding	Used LabelEncoder, OrdinalEncoder, OneHotEncoder, and custom GetDummies transformer
Create new columns	No	Dummy columns created via transformation; not counted as new columns
Feature selection	Yes	Top features selected based on XGBoost feature importance (`viz.features_`)
Data scaling/standardisation	Yes	`MinMaxScaler` applied to encoded features before modeling
Hyperparameter tuning	Yes	Extensive `GridSearchCV` used for Logistic Regression, Random Forest, and XGBoost

5th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Target (`Credit_Score`) class distribution checked via `value_counts(normalize=True)` and displayed across train/val/test.
Sampling type	Random	Used `train_test_split()` with fixed random state, without `stratify`.
Outliers removal	No	No explicit filtering, trimming, or IQR/Tukey-based removal of outliers was performed.
Check for duplicates	No	No `.duplicated()` or `drop_duplicates()` check observed.
Imputation of missing values	None	No imputation observed; clean dataset assumed from earlier cleaning phase (not repeated in this notebook).
Drop columns	Yes	Columns `ID`, `Name`, `Customer_ID` are dropped.

Encoding	mixture of encoding	Used `LabelEncoder` for target, `OrdinalEncoder` for categorical features, `GetDummies` for multi-hot, and `OneHotEncoder` for object variables.
Create new columns	No	Columns derived from parsing (e.g., `GetDummies`, `OneHotEncoder`) or encoded, but no brand new features introduced.
Feature selection	Yes	Used model-based (XGBoost + yellowbrick + permutation) and correlation methods to drop features and select final 9 columns.
Data scaling/standardisation	Yes	Applied `MinMaxScaler` on final processed datasets.
Hyperparameter tuning	Yes	Performed `GridSearchCV` for Logistic Regression, Random Forest, and XGBoost with multiple parameters and scoring metrics (recall-focused).

Ground Truth

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	`df_train["Credit_Score"].value_counts(normalize=True)` printed to inspect target balance.
Sampling type	Random	`train_test_split(X, y, test_size=0.2, random_state=42)` with no `stratify`.
Outliers removal	No	No IQR/quantile/rule-based filtering found.
Check for duplicates	No	No `.duplicated()` / `.drop_duplicates()` used.
Imputation of missing values	Ignore	No explicit imputation; modelling proceeds on cleaned data (encoders handle unknowns, not NAs).
Drop columns	Yes	Dropped identifiers `['ID','Customer_ID','Month','Name','SSN']` without reuse.

Encoding	Mixture	Custom `GetDummies` for multi-valued strings → `OneHotEncoder` for remaining categoricals (LogReg path), and `OrdinalEncoder` for tree/XGB path; `LabelEncoder` for target.
Create new columns	No	One-hot/dummy expansions are derived from existing columns; not counted as “new”.
Feature selection	Yes	Model-based: XGBoost feature importances / permutation importances used to pick top features; final model trained on selected subset.
Data scaling/standardisation	Yes	`MinMaxScaler()` applied on the one-hot encoded feature set for Logistic Regression.
Hyperparameter tuning	Yes	`GridSearchCV` for Logistic Regression, RandomForest, and XGBoost with specified grids and 5-fold CV.

RECIPE 10
1st Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used `.value_counts(normalize=True)` on `credit_score` to inspect class proportions.
Sampling type	Random	Used `train_test_split(test_size=0.25, random_state=42)` — random sampling.
Outliers removal	Yes	Applied IQR-based outlier detection on numeric features and dropped multiple outlier rows.
Check for duplicates	Yes	Used `data.duplicated().sum()` — confirmed no duplicates.
Imputation of missing values	Mixture of imputation	Used random sampling for numeric and categorical columns; replaced invalid strings with `np.nan`.
Drop columns	Yes	Dropped irrelevant columns: ID, Customer_ID, SSN, name, credit_mix, credit_utilization_ratio, etc.

Encoding	Mixture of encoding	OneHotEncoding for categorical features, LabelEncoding for target variable, manual multi-label binarization for `type_of_loan`.
Create new columns	Yes	Created 8 binary columns from `type_of_loan` multi-label string field.
Feature selection	No	No explicit technique like correlation filtering or model-based feature importance observed.
Data scaling or standardisation	No	Defined scalers but did not apply them to the data.
Hyperparameter tuning	No	Manual changes to model params (e.g., max_depth, n_estimators) but no GridSearchCV or similar technique.

2nd Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Class balance was visualized using `value_counts()` and `countplot()` on `Credit_Score`.
Sampling type	Random	`train_test_split()` used without `stratify` or oversampling.
Outliers removal	Yes	IQR method applied to numerical columns (`detect_outliers()`), rows with >2 outliers were removed.
Check for duplicates	Yes	`data.duplicated().sum()` used to identify duplicates.
Imputation of missing values	Mixture of imputation techniques	Includes random selection from non-missing values (categorical/numeric), string replacement (`NM`, `'__10000__'`), numeric mean imputation, etc.
Drop columns	Yes	Columns like `ID`, `Name`, `SSN`, `Type_of_Loan` were dropped with no replacement.
Encoding	Mixture of encoding	LabelEncoder (target), OneHotEncoder, and manual dummy variable creation from multi-valued strings.

Create new columns	Yes	New columns created from `Type_of_Loan` parsing (e.g., `payday_loan`, `student_loan`, etc.).
Feature selection	Yes	Features removed due to correlation, visualization, or post-EDA reasoning. Also, feature importance used in some notebooks.
Data scaling or standardisation	Yes	`MinMaxScaler`, `StandardScaler`, and `RobustScaler` used in different notebooks.
Hyperparameter tuning	Yes	`GridSearchCV` used for Logistic Regression, Random Forest, and XGBoost models with scoring (recall) and parameter grids.

3rd Prompt
Accuracy 10/11

Step	Used?	Details
Check for balanced data	Yes	Target distribution + plots
Sampling type	Random	`train_test_split()`
Outlier removal	Yes	IQR-based function
Check for duplicates	Yes	`data.duplicated()`
Missing value imputation	Yes	Custom random fill for all missing
Drop columns	Yes	IDs, names, etc. dropped
Encoding	Label, OneHot	Both used explicitly
Create new columns	Yes	Dummy variables from `type_of_loan`
Feature selection	Partial	Manual pruning only
Standardization	No	Not applied to features
Hyperparameter tuning	No	Manual changes only

4th Prompt
Accuracy 11/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Class distribution of `credit_score` checked using `.value_counts(normalize=True)` and plots
Sampling type	Random	`train_test_split(..., random_state=42)` used without `stratify=`
Outliers removal	Yes	IQR-based outlier detection and dropping via `detect_outliers()`
Check for duplicates	Yes	`data.duplicated().sum()` used to check for duplicates
Imputation of missing values	mixture of imputation techniques	Categorical and numeric columns filled using random choice, hardcoded values, and conversion
Drop columns	Yes	Dropped columns like `id`, `customer_id`, `name`, `ssn`, `credit_mix`, etc.
Encoding	mixture of encoding	One-hot encoding + manual multi-hot dummy creation + LabelEncoder for target
Create new columns	No	All new columns (e.g., loan types) derived from `type_of_loan`
Feature selection	Yes	Dropping columns post EDA and correlation
Data scaling/standardisation	No	No `StandardScaler`, `MinMaxScaler`, or similar used
Hyperparameter tuning	No	Multiple models evaluated, but all parameters were fixed manually

5th Prompt
Accuracy 9/11

Data Wrangling Step	Technique Used	Details
Check for balanced data	Yes	Used `value_counts()` and `factorplot()` on `credit_score` to observe class distribution.

Sampling type	Random	Used `train_test_split()` without stratify, so sampling is random.
Outliers removal	Yes	Applied IQR-based method to detect and drop outliers across all numeric columns using `detect_outliers()` function.
Check for duplicates	Yes	Checked using `data.duplicated().sum()` early in the notebook.
Imputation of missing values	mixture of imputation techniques	Numeric: `np.random.choice()` imputation; Categorical: `np.random.choice()` + row drops + string replacements.
Drop columns	No	Dropped multiple columns (`ssn`, `credit_mix`, `type_of_loan`, etc.) after EDA and parsing; some used for feature engineering, so counted under FS.
Encoding	mixture of encoding	Used `get_dummies`, `LabelEncoder`, and manual parsing for `type_of_loan`.
Create new columns	Yes	Created binary columns (`auto_loan`, `payday_loan`, etc.) from multi-valued `type_of_loan` column.
Feature selection	Yes	Dropped features based on low variance, EDA insights, and correlation heatmaps.
Data scaling/standardisation	No	No use of `StandardScaler`, `MinMaxScaler`, or `RobustScaler`.
Hyperparameter tuning	No	All models trained with fixed parameters (`max_depth`, `n_estimators`); no use of `GridSearchCV`, `RandomizedSearchCV`, or similar.

Ground Truth

Data Wrangling Step	Technique Used	Details

Check for balanced data	Yes	`data.credit_score.value_counts()` used to inspect target distribution.
Sampling type	Random	`train_test_split(X, Y, test_size=0.25, random_state=42)` without `stratify`.
Outliers removal	Yes	IQR-based detector `detect_outliers(...)`; identified indices dropped from `data`.
Check for duplicates	Yes	Duplicates checked with `data.duplicated().sum()`.
Imputation of missing values	Mixture of imputation techniques	Random sampling imputation for numeric (`random_selection_filling_missing_values_2`) and categorical (`random_selection_filling_missing_values_cat`) columns.
Drop columns	Yes	Dropped without replacement: `['id','customer_id','name','ssn','credit_mix','credit_utilization_ratio','payment_behaviour','num_bank_accounts','num_credit_card']`. (Drop of `type_of_loan` not counted here as it's replaced by indicators.)
Encoding	Mixture	Manual one-hot style indicators for `type_of_loan`; `pd.get_dummies` on `['month','occupation','payment_of_min_amount']`; `LabelEncoder` for target `credit_score`.
Create new columns	No	Indicator/dummy columns are derived from existing fields; not counted as "new".
Feature selection	Yes	Columns removed after EDA/processing (post-visualisation drop counts as feature selection).
Data scaling/standardisation	No	No `StandardScaler`/`MinMaxScaler` applied before modelling.
Hyperparameter tuning	No	Multiple fixed trials (KNN k=1/2/6/15; RF n_estimators=25/50/75/100) without systematic search (`GridSearchCV`/`RandomizedSearchCV`).