This is the flowchart link please visit with 100 % zoom for a better view

<https://www.canva.com/design/DAGIEfZwcqU/ctyxiBGl_wpmNkB2vRlkqg/edit?utm_content=DAGIEfZwcqU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

**project explanation**

1. **Load Data**:

* Load the training and prediction data from CSV files in chunks.
* This step involves loading the data in chunks due to memory constraints, and then concatenating these chunks into a single DataFrame for both training and prediction datasets.

1. **Initial Data Inspection**:

* Inspect the loaded data for information like column types, missing values, etc.
* This step helps understand the structure of the data and identify columns with missing values or other issues that need to be addressed.

1. **Handle Missing Values**:

* Impute missing values using appropriate strategies for different types of columns (numerical, categorical, ordinal, binary).
* Missing values can disrupt the modeling process. Thus, imputation is necessary to handle them appropriately.

1. **Binary Variable Handling**:

* For binary variables, replace values greater than 1 with 1.
* This ensures that binary variables have only 0 or 1 values, maintaining consistency.

1. **Correlation Analysis**:

* Calculate and visualize the correlation matrix.
* Identifying highly correlated features helps understand the relationships between variables and decide on feature selection or engineering steps.

1. **Outlier Handling**:

* Apply Winsorization to numerical columns to handle outliers.
* Winsorization limits extreme values, reducing their impact on the model without removing any data.

1. **Normalize Numerical Features**:

* Apply StandardScaler to normalize numerical columns.
* Normalizing numerical features ensures that they have a similar scale, which helps improve model performance.

1. **Categorical Data Conversion**:

* Convert categorical columns to 'category' data type.
* This step is necessary for models like LightGBM that can handle categorical features directly.

1. **Feature Selection and Splitting Data**:

* Select features for training, and split the data into training and validation sets.
* Preparing the dataset by separating features and the target variable, then splitting into training and validation sets helps in model evaluation.

1. **Class Imbalance Handling**:

* Calculate class weights to handle imbalanced classes.
* Adjusting for class imbalance ensures that the model does not become biased towards the majority class.

1. **Train LightGBM Model**:

* Train the LightGBM classifier using the training data.
* LightGBM is a powerful model for classification tasks. Training involves fitting the model to the data.

1. **Model Evaluation**:

* Predict on validation set and evaluate using F1 score.
* The F1 score provides a balance between precision and recall, which is important for imbalanced datasets.

1. **Prepare Prediction Data**:

* Apply the same preprocessing steps to the prediction data as done on training data.
* Consistent preprocessing ensures that the model can correctly interpret the prediction data.

1. **Make Predictions**:

* Use the trained model to make predictions on the prediction data.
* Generate predictions for the new data.

1. **Save Predictions**:
2. Save the predictions to a CSV file for submission or further analysis.
3. This step outputs the final results in the required format.
4. Save the trained model for future use.
5. **Hyperparameter Tuning**

Hyperparameter tuning was attempted using RandomizedSearchCV and Optuna libraries. However, due to memory constraints and limited computational resources, the tuning process could not be completed successfully. The tuned models could not be loaded after an hour of runtime.

**Conclusion**

The project successfully built a LightGBM model for the binary classification problem, addressing various data preprocessing challenges and handling class imbalance. While hyperparameter tuning could not be fully executed, the model's performance on the validation set was reported. The project used the machine learning techniques, including data preprocessing, exploratory data analysis, model training, evaluation, and prediction.