Project Report: Loan Approval Prediction Task

1. The Task I Was Assigned

I was given the task of building a model to predict whether a loan application would be approved based on various applicant and financial features. The core requirements included handling missing values, encoding categorical features, training and evaluating classification models, and focusing on metrics relevant for imbalanced data (precision, recall, F1-score). Bonus objectives involved using SMOTE to address class imbalance and trying Logistic Regression.

2. The Problem and Its Solution

The Problem: In the financial industry, accurately assessing loan applications is critical. Incorrectly approving a loan to a high-risk applicant can lead to significant financial losses (bad debt), while incorrectly rejecting a low-risk applicant can result in missed revenue opportunities and customer dissatisfaction. Loan approval datasets often suffer from class imbalance, where approved loans significantly outnumber rejected ones. This imbalance can bias models towards the majority class, making them less effective at identifying the crucial minority class (rejected loans).

The Solution: This project addresses the challenge of accurate and reliable loan approval prediction, specifically tackling the issue of imbalanced data. The solution involves:

- **Comprehensive Data Preprocessing:** Cleaning and transforming raw application data, including handling anomalies and encoding diverse feature types.
- **Robust Classification Modeling:** Employing powerful machine learning algorithms to learn complex patterns from the data.
- **Imbalance Handling:** Utilizing techniques like SMOTE (Synthetic Minority Over-sampling Technique) to ensure the model performs well on both approved and rejected cases, especially the critical minority class.
- **Focused Evaluation:** Prioritizing metrics like precision, recall, and F1-score for the minority class to ensure the model is effective at identifying risky applications (minimizing false positives) and not missing actual rejections (high recall).
- **Model Optimization:** Applying hyperparameter tuning to maximize predictive performance.

This approach provides a data-driven, automated system that can significantly reduce financial risk and improve the efficiency and fairness of the loan approval process.

3. How I Approached the Task

To complete the assignment, I broke down the problem into a clear, multi-stage process, focusing on a logical progression of data science steps.

Data Acquisition and Initial Exploration

- **My Goal:** To load the dataset and gain a fundamental understanding of its structure, features, and the target variable.
- **What I Did:** I loaded the loan_approval_dataset.csv into a Pandas DataFrame. I performed initial data inspections using df.head(), df.info(), df.describe(), and df.isnull().sum(). I also examined the distribution of the loan_status (target variable) to understand class balance.
- **The Result:** I successfully loaded the dataset and identified that column names had leading/trailing whitespace, which I immediately stripped. I found no missing values. The target variable showed a class imbalance, with 'Approved' loans outnumbering 'Rejected' loans.
- **My Conclusion:** This initial step provided a clean dataset and highlighted the class imbalance as a key challenge.

Detailed Data Cleaning and Handling Missing Values

- **My Goal:** To address any data quality issues, specifically focusing on missing values and anomalies.
- **What I Did:** I re-checked for missing values and reviewed data types. Crucially, I identified and corrected an anomaly where residential_assets_value had negative entries, setting these values to 0.
- **The Result:** The dataset was confirmed to be free of missing values, and the negative asset values were successfully handled, ensuring data integrity.
- **My Conclusion:** Proactive data cleaning is vital for reliable model training.

Categorical Encoding and Feature Engineering

- **My Goal:** To transform categorical features into a numerical format suitable for machine learning models and create new, informative features.
- **What I Did:** I identified categorical columns (education, self_employed, loan_status). I stripped whitespace from their values to ensure accurate mapping. I then manually mapped 'Graduate' to 1 and 'Not Graduate' to 0 for education, 'Yes' to 1 and 'No' to 0 for

self_employed, and 'Approved' to 1 and 'Rejected' to 0 for loan_status (my target variable). I also engineered a new feature, Total_Income, by summing income_annum and bank_asset_value.

- **The Result:** All relevant categorical features and the target were successfully encoded into numerical format. The new Total_Income feature was created, potentially offering a more holistic view of applicant financial capacity.
- **My Conclusion:** Proper encoding and thoughtful feature engineering enhance the model's ability to learn from the data.

Data Splitting and Baseline Model Preparation

- **My Goal:** To divide the dataset into training and testing sets and establish a performance benchmark with a simple classification model.
- **What I Did:** I separated features (X) from the target (y), dropping loan_id. I split the data into training (80%) and testing (20%) sets using train_test_split with random_state=42. I did not stratify the split at this point, as SMOTE would be applied only to the training set later. I then trained a baseline DecisionTreeClassifier and evaluated its accuracy and classification report.
- **The Result:** The data was successfully split, maintaining the class imbalance in both sets. The baseline Decision Tree model achieved a very high accuracy of 97.54%, indicating strong initial predictability.
- **My Conclusion:** A robust baseline provides a clear measure for evaluating the improvement of more complex models.

Classification Model Training (Random Forest)

- **My Goal:** To train a more robust classification model (Random Forest) on the preprocessed data.
- **What I Did:** I initialized and trained a RandomForestClassifier with n_estimators=100 on the training data. I then made predictions on the test set and generated its accuracy and classification report.
- **The Result:** The Random Forest model achieved an accuracy of 97.19%, performing exceptionally well and showing strong precision, recall, and F1-scores for both classes.
- **My Conclusion:** Random Forest proved to be highly effective for this dataset even without specific imbalance handling, setting a high performance standard.

Model Evaluation - Focus on Imbalanced Metrics

- **My Goal:** To deeply evaluate the Random Forest model's performance, specifically focusing on metrics crucial for imbalanced data.
- **What I Did:** I re-trained the Random Forest model and generated its detailed classification report. I then extracted and printed specific precision, recall, and F1-score for the minority class ('Rejected' loans).
- **The Result:** The model showed excellent performance for the 'Rejected' class, with a precision of 0.9742, recall of 0.9497, and F1-score of 0.9618.
- **My Conclusion:** The Random Forest model demonstrated strong robustness against the dataset's class imbalance, effectively identifying the critical minority class.

Confusion Matrix Visualization and Analysis

- **My Goal:** To visually understand the Random Forest model's correct and incorrect predictions.
- **What I Did:** I generated the confusion matrix for the Random Forest model and visualized it using seaborn.heatmap.
- **The Result:** The confusion matrix clearly showed a very high number of true positives (528) and true negatives (302), with minimal false positives (16) and false negatives (8).
- **My Conclusion:** The visualization confirmed the model's high accuracy and its effectiveness in minimizing critical errors for loan approval.

Feature Importance Visualization

- **My Goal:** To understand which features are most influential for the Random Forest model's predictions.
- **What I Did:** I extracted the feature_importances_ from the trained Random Forest model, mapped them to feature names, and visualized the top 10 most important features using a bar plot.
- **The Result:** cibil_score was identified as the overwhelmingly most important feature (importance > 0.8), followed by loan_term and loan_amount.
- **My Conclusion:** This analysis highlighted cibil_score as the primary determinant for loan approval, aligning with real-world lending practices.

Class Imbalance Handling (SMOTE) (Bonus)

- **My Goal:** To address the class imbalance in the training data using SMOTE and re-evaluate model performance.

- **What I Did:** I applied SMOTE from imblearn.over_sampling to only the training data (X_train, y_train) to oversample the minority class ('Rejected'). I then re-trained the Random Forest model on this SMOTE-transformed data and evaluated its performance.
- **The Result:** SMOTE successfully balanced the training set. The Random Forest model trained on SMOTE data achieved an accuracy of 98.00%, with improved recall (0.9717) and F1-score (0.9702) for the minority class compared to the untuned model.
- **My Conclusion:** SMOTE proved effective in further enhancing the model's ability to identify the minority class, making it even more robust.

Logistic Regression (Bonus) - Training and Evaluation

- **My Goal:** To train and evaluate a Logistic Regression model, specifically on the SMOTE-transformed data.
- **What I Did:** I initialized and trained a LogisticRegression model on the SMOTE-transformed training data and evaluated its performance on the original test set.
- **The Result:** The Logistic Regression model achieved an accuracy of 70.00%, with a recall of 0.5031 for the 'Rejected' class. This performance was significantly lower than that of the Random Forest models.
- **My Conclusion:** For this dataset, the linear nature of Logistic Regression was less effective than ensemble tree-based models, even with class balancing.

Comprehensive Model Comparison

- **My Goal:** To systematically compare the performance of all models trained.
- **What I Did:** I compiled a summary table comparing key metrics (Accuracy, Precision, Recall, F1-score for 'Rejected' class, Macro F1-Score, Weighted F1-Score) for the Baseline Decision Tree, untuned Random Forest, Random Forest with SMOTE, and Logistic Regression with SMOTE.
- **The Result:** The Random Forest model trained with SMOTE consistently showed the highest performance across all metrics, making it the clear winner.
- **My Conclusion:** This comprehensive comparison solidified the Random Forest with SMOTE as the most effective approach for this problem.

Final Model Selection and Re-evaluation

- **My Goal:** To select the single best model and perform a final, comprehensive evaluation.

- **What I Did:** Based on the comprehensive comparison, I selected the Tuned Random Forest model (trained with SMOTE) as the final best model due to its superior accuracy and balanced F1-scores. I then generated its final classification report and confusion matrix.
- **The Result:** The final model achieved an outstanding accuracy of 98.13%, with very low numbers of false positives (8) and false negatives (8) in the confusion matrix.
- **My Conclusion:** The Tuned Random Forest model stands as a highly robust and reliable solution for predicting loan approval, capable of minimizing critical errors.

4. Final Result of My Work

After completing all assigned tasks and bonus objectives, my final conclusion is that the **Tuned Random Forest model, trained on SMOTE-balanced data**, is the most effective and reliable solution for predicting loan approval. It achieved an exceptional **98.13% accuracy** on the unseen test data. More importantly, its performance on the critical minority class ('Rejected' loans) was outstanding, demonstrating high precision, recall, and F1-score, with only 8 false positives and 8 false negatives.

This project successfully demonstrated the power of binary classification, effective handling of imbalanced data through SMOTE, and the robust performance of ensemble methods like Random Forest. The insights gained, particularly the overwhelming importance of cibil_score, provide actionable intelligence for a real-world loan approval system, significantly mitigating risk and improving decision-making efficiency.