**Credit Card Default Prediction: Enabling Banks to Mitigate Financial Risks**

During my tenure with the Society of Business at IIT Roorkee, I worked on a predictive analytics project aimed at identifying potential credit card defaulters. The objective was to help financial institutions proactively mitigate the risk of defaults, which can result in significant financial losses. We worked with a dataset of **30,000 rows and 26 columns**, encompassing customer demographic data and six months of historical payment records (April to September).

**1. Problem Identification**

Credit card defaults pose a critical challenge for banks, necessitating accurate prediction models. The key challenge was to design a model that balanced **precision** (minimizing false positives) and **recall** (minimizing false negatives). This balance is essential for reducing unnecessary penalties on customers while ensuring that potential defaulters are flagged effectively. The overarching goal was to provide a reliable tool for financial institutions to act proactively in mitigating risks.

**2. Data Cleaning and Preprocessing**

We began by addressing data quality issues:

* **Handling Missing Values:**  
  The dataset had less than **1% null values**. Given the dataset's size, removing these rows was a practical solution, as it didn’t significantly affect the analysis.
* **Feature Engineering:**  
  To enhance the dataset’s predictive power, I engineered the following features:
  1. **Credit Utilization Ratio:** The ratio of the amount used to the credit limit, indicating financial strain.
  2. **Payment-to-Bill Ratio:** The ratio of payments made to the billed amount, reflecting repayment behavior.
  3. **Past Payments Consistency:** A measure of consistency in payments over six months, providing insights into the customer’s payment behavior.

These features significantly improved the model’s ability to capture subtle patterns in the data.

**3. Data Transformation**

To prepare the dataset for modeling, I performed the following transformations:

1. **Standardization:**  
   Features such as bill amounts and past payments were standardized to ensure all numerical variables were on a comparable scale. This step was crucial for algorithms like logistic regression, which are sensitive to feature scales.
2. **One-Hot Encoding:**  
   Categorical variables, such as education level and marital status, were converted into numerical formats using one-hot encoding. This prevented the model from misinterpreting the ordinal nature of these features.
3. **Principal Component Analysis (PCA):**  
   To reduce dimensionality and address multicollinearity (e.g., six months of bill amounts and payments being highly correlated), I applied PCA. By retaining **95% of the variance**, we reduced the dataset to a smaller set of components, improving computational efficiency and reducing the risk of overfitting.

**4. Model Development**

I implemented and compared multiple machine learning models to find the most effective solution:

1. **Logistic Regression:**  
   Served as a baseline model to set initial benchmarks.
2. **Decision Tree:**  
   Provided interpretability by visualizing decision paths.
3. **Random Forest:**  
   Improved accuracy through ensemble learning, aggregating predictions from multiple decision trees.
4. **XGBoost:**  
   Leveraged advanced gradient boosting techniques for optimal performance.

**5. Model Evaluation**

To evaluate model performance, I prioritized the **F1-score** as the primary metric. This metric balances **precision** and **recall**, addressing the project’s dual objectives:

* **Precision:** To avoid penalizing customers who would pay on time (minimizing false positives).
* **Recall:** To ensure all potential defaulters were flagged (minimizing false negatives).

Among the models, **XGBoost** delivered the best performance:

* **Precision:** 85%
* **Recall:** 82%  
  This balance made the model effective for real-world deployment.

**6. Results and Insights**

Key insights from the project included:

1. Customers with **high credit utilization ratios** and **low payment-to-bill ratios** were more likely to default, indicating financial strain or repayment difficulties.
2. **Consistency in past payments** emerged as a strong predictor of default behavior. Customers who missed payments consistently over six months had a significantly higher likelihood of defaulting.

**7. Challenges and Solutions**

The primary challenge was addressing the **class imbalance** in the dataset, with far fewer defaults compared to non-defaults. This imbalance could lead to biased models. To resolve this:

* I employed **SMOTE (Synthetic Minority Oversampling Technique)** to generate synthetic samples for the minority class. This method effectively balanced the dataset, improving the **recall score** without significantly compromising precision.

**8. Reflection and Takeaways**

This project provided valuable learnings in multiple areas:

1. **Feature Engineering:**  
   The creation of features like credit utilization and payment consistency demonstrated how domain knowledge can enhance model performance.
2. **Dimensionality Reduction with PCA:**  
   PCA proved invaluable in handling correlated features and reducing computational complexity without sacrificing critical information.
3. **Model Optimization:**  
   Understanding the trade-offs between precision and recall helped me design models that aligned with real-world business needs.
4. **Balanced Metrics for Decision-Making:**  
   By prioritizing metrics like the F1-score, I learned how to make data-driven decisions that balance competing objectives.

I am eager to apply these skills in future projects, leveraging data science to drive impactful business outcomes.

**Project Description**

* **Business Objective**: Banks wanted to reduce financial losses by proactively identifying customers at risk of defaulting on credit payments. This not only protects the bank's bottom line but also maintains customer relationships by potentially restructuring loans or offering tailored solutions.
* **Scope of Application**: The model could assist in:
  + Setting credit limits.
  + Identifying high-risk customers for counselling or proactive outreach.
  + Enhancing risk-adjusted interest rates.

**Technical and Methodological Details**

1. **Data Overview**:
   * **Dataset Source**: UCI Machine Learning Repository – Defaults of Credit Cards.
   * **Distribution**:
     + Default rate: What percentage of customers defaulted in the dataset? - 77.88% (23,364 instances).
     + Non-default rate: What percentage of customers did not default? - 22.12% (6,636 instances).
   * **Class Imbalance**:
     + Imbalance Ratio: ~3.52:1 (Non-Defaults: Defaults)
2. **Data Cleaning**:
   * Boxplots, IQR, or Z-Score used to detect extreme values in "bill amounts" and "payment amounts."
   * Outliers in features like "bill amounts" or "payment amounts" were treated (e.g., Winsorizing, capping, or removal).
   * Handling incorrect data entries (e.g., negative payments or bill amounts).
3. **Feature Engineering**:
   * Checked for multicollinearity using Variance Inflation Factor (VIF) before applying PCA.
   * A VIF > 5 indicated high correlation between features.
   * Ensured the new features (like payment consistency) added value to the model and weren't redundant by verifying model performance with and without the new features.
4. **PCA Details**:
   * Number of Components Retained: Selected components explaining 95% of the variance (~10 components).
   * Implementation: Decided threshold manually after plotting the cumulative explained variance curve and ensuring sufficient information retention.
5. **Model Development**:
   * Did you use a validation technique like **k-fold cross-validation** to ensure robustness?
   * Hyperparameter tuning:
     + What tuning techniques were used (e.g., Grid Search, Random Search, Bayesian Optimization)?
     + Example of key parameters tuned for XGBoost (e.g., learning rate, max\_depth, n\_estimators).
6. **Evaluation Metrics**:
   * Mention other metrics you tracked (e.g., ROC-AUC, accuracy) and why F1-score was prioritized over these.
   * Confusion matrix details (e.g., number of true positives, false positives, true negatives, and false negatives).
7. **SMOTE Details**:
   * The specific oversampling ratio applied.
   * How did you evaluate the impact of SMOTE on the model’s performance before and after implementation?

**Business Outcomes**

* **Deployment Potential**: Discuss how this project could integrate into a bank’s workflow:
  + As a real-time monitoring tool.
  + In periodic customer risk assessment reports.
  + Combined with expert domain knowledge for holistic decision-making.
* **Cost-Benefit Analysis**:
  + Quantify potential savings by reducing default rates by a certain percentage.
  + Address how the cost of false positives could impact customer experience and operational costs.

**Challenges and Learning**

1. **Handling Imbalance**:
   * Did you try alternative methods like class-weighted loss functions, under-sampling, or ensemble balancing techniques?
   * How did you ensure the SMOTE-synthesized samples were representative of real-world cases?
2. **Computational Challenges**:
   * Mention any issues with computation time or memory due to the large dataset.
   * Tools or strategies used to optimize these challenges (e.g., feature selection or distributed computing tools like Dask or Spark).
3. **Interpreting the Model**:
   * Techniques used to interpret the XGBoost model (e.g., SHAP values, feature importance scores).
   * How insights from the model were validated with domain knowledge.
4. **Feature Engineering Validation**:
   * How did you validate that the engineered features improved the model's predictive power? (e.g., using feature importance rankings or cross-validation results).

**Further Improvements**

1. **Algorithm**:
   * Could deep learning (e.g., neural networks) or time-series analysis (given the 6-month data) be explored?
   * Implementing advanced boosting algorithms like CatBoost or LightGBM for comparison.
2. **Data Augmentation**:
   * Use synthetic datasets to model out-of-distribution scenarios.
3. **Real-World Application**:
   * Enhancing the model to work with real-time data pipelines using tools like Apache Kafka or AWS Kinesis.
   * Incorporating real-time decision thresholds and model retraining.

**Common Interview Questions**

1. **Business Understanding**:
   * How would you convince a non-technical stakeholder of the model's reliability?
   * What measures can the bank take apart from predictive modeling to reduce defaults?
2. **Data Challenges**:
   * If null values were more significant, what strategies would you have adopted?
   * How would you handle non-stationarity in customer behavior over time?
3. **Modeling**:
   * Why did XGBoost outperform Random Forest in this scenario?
   * How did you address potential overfitting, especially in a high-dimensional dataset?
4. **Performance**:
   * How would you deploy this model in a low-latency environment?
   * If precision had been prioritized over recall, what would be the trade-offs, and how would you modify the model?
5. **Real-World Adaptation**:
   * What are the ethical considerations of using such models for financial decision-making?
   * How would you adapt this model to handle new customer demographics or economic shifts?

**Reflection**

* How you would approach a similar project differently now, based on this experience:
  + E.g., integrating time-series analysis, deploying models for real-time risk detection, or automating feature engineering processes.