Credit Card Cross-Selling Analysis And Prediction

Introduction:

Project Overview

- **Context:** The project targets predicting potential credit card leads for Happy Customer Bank.
- The bank aims to identify customers with a higher likelihood of taking a recommended credit card. Various customer demographics, account details, and engagement metrics are included to support targeted marketing.
- **Dataset Summary:** Key features include Age, Average Account Balance, Occupation, and Credit Product status, with the target variable being Is_Lead.
- **Business Importance:** Accurate prediction can enhance marketing efficiency and customer targeting

Objectives:

Project Objectives

1. **Data Understanding & Cleaning:**
    - Handle missing values.
    - Perform categorical encoding and feature scaling.
2. **Exploratory Data Analysis (EDA):**
    - Analyze customer demographics and financial behavior.
    - Visualize relationships between features and the target variable.
3. **Model Development & Evaluation:**
    - Test and compare different machine learning models.
    - Tune hyperparameters for better performance.
4. **Model Selection:**
    - Choose the best model based on precision, recall, and F1-score.

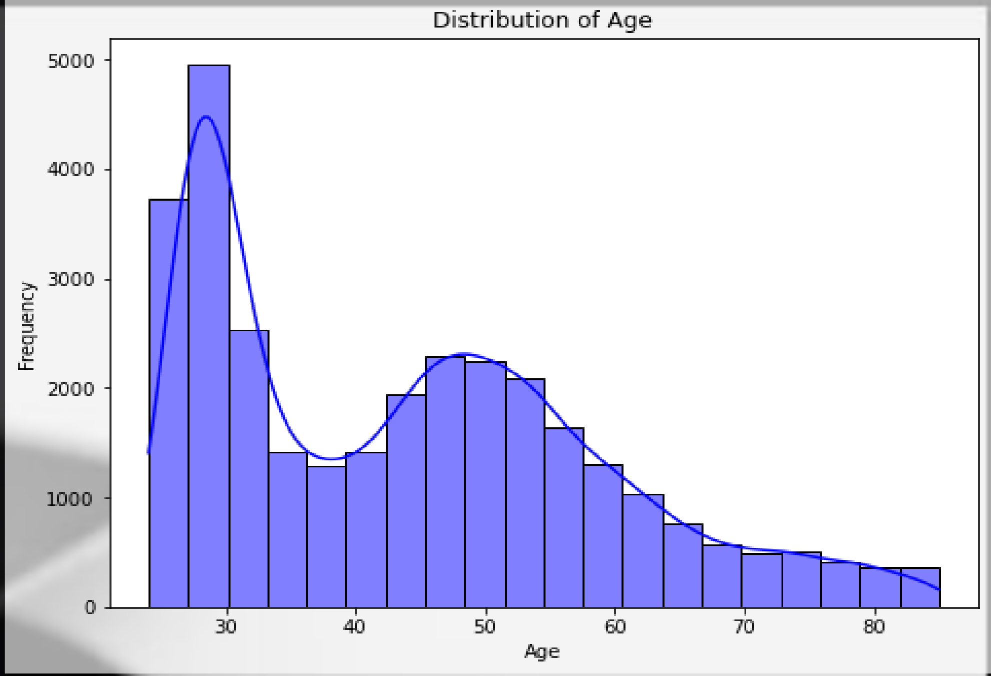
Observations:

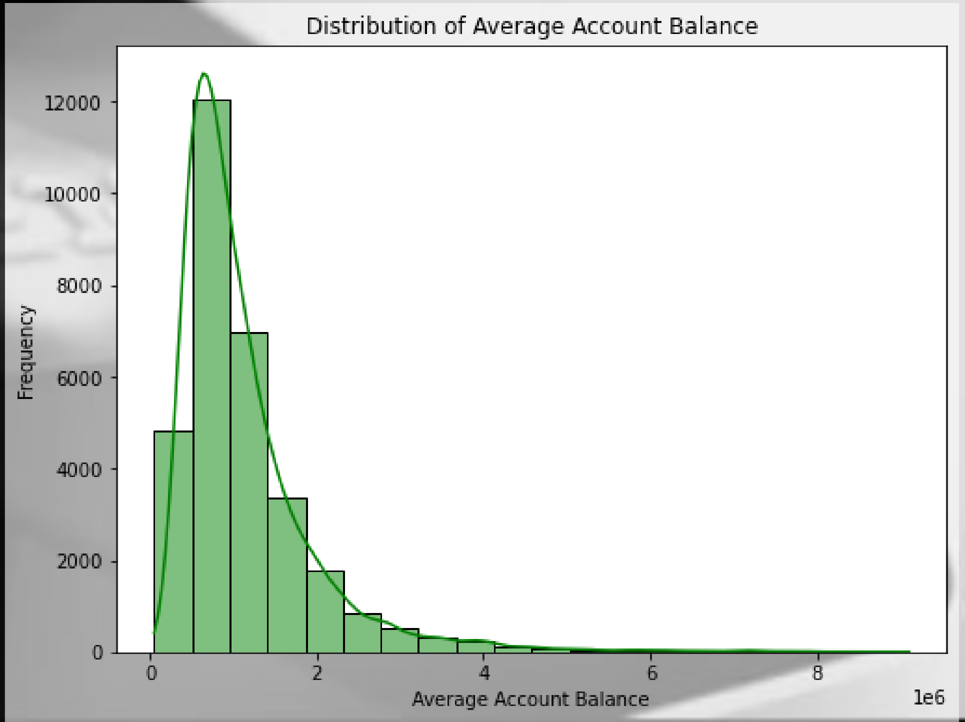
The Age distribution is likely unimodal, with most values concentrated around a specific range (e.g., 25–45 years), depending on the dataset.

Skewness:

Since the distribution is right-skewed it indicates younger customers dominate.

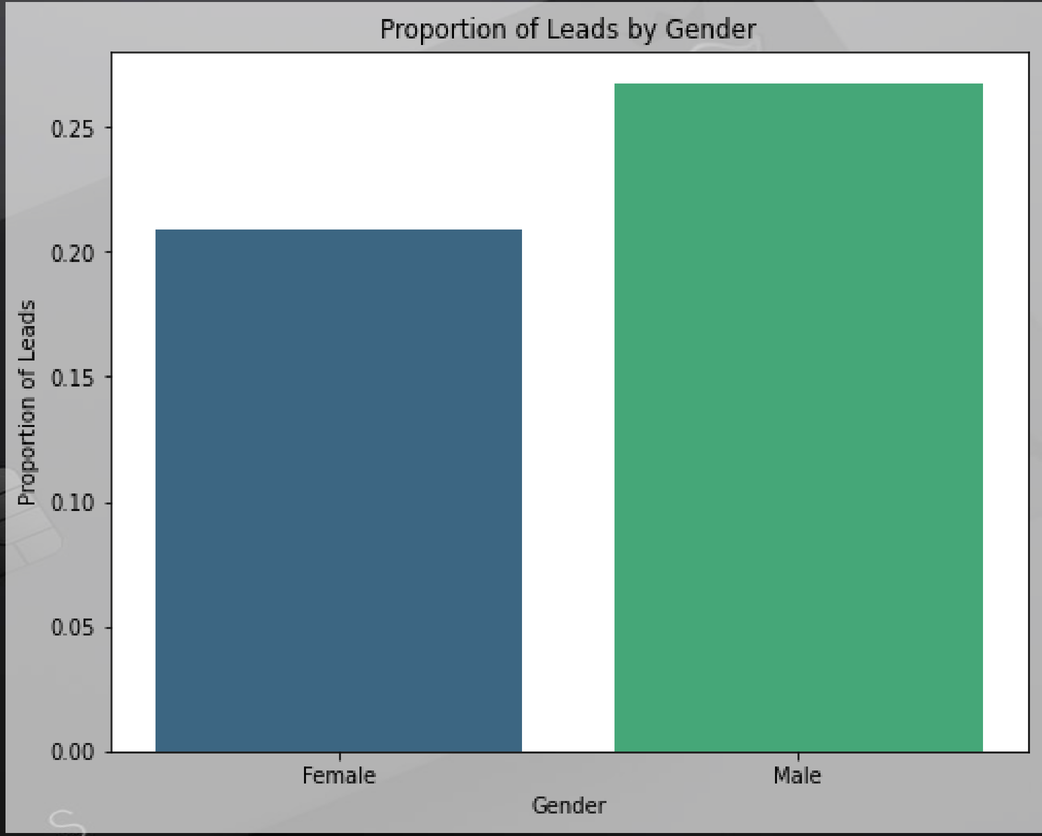
Avg_Account_Balance might show significant right-skewness, meaning most customers have a low balance while a few have very high balances.



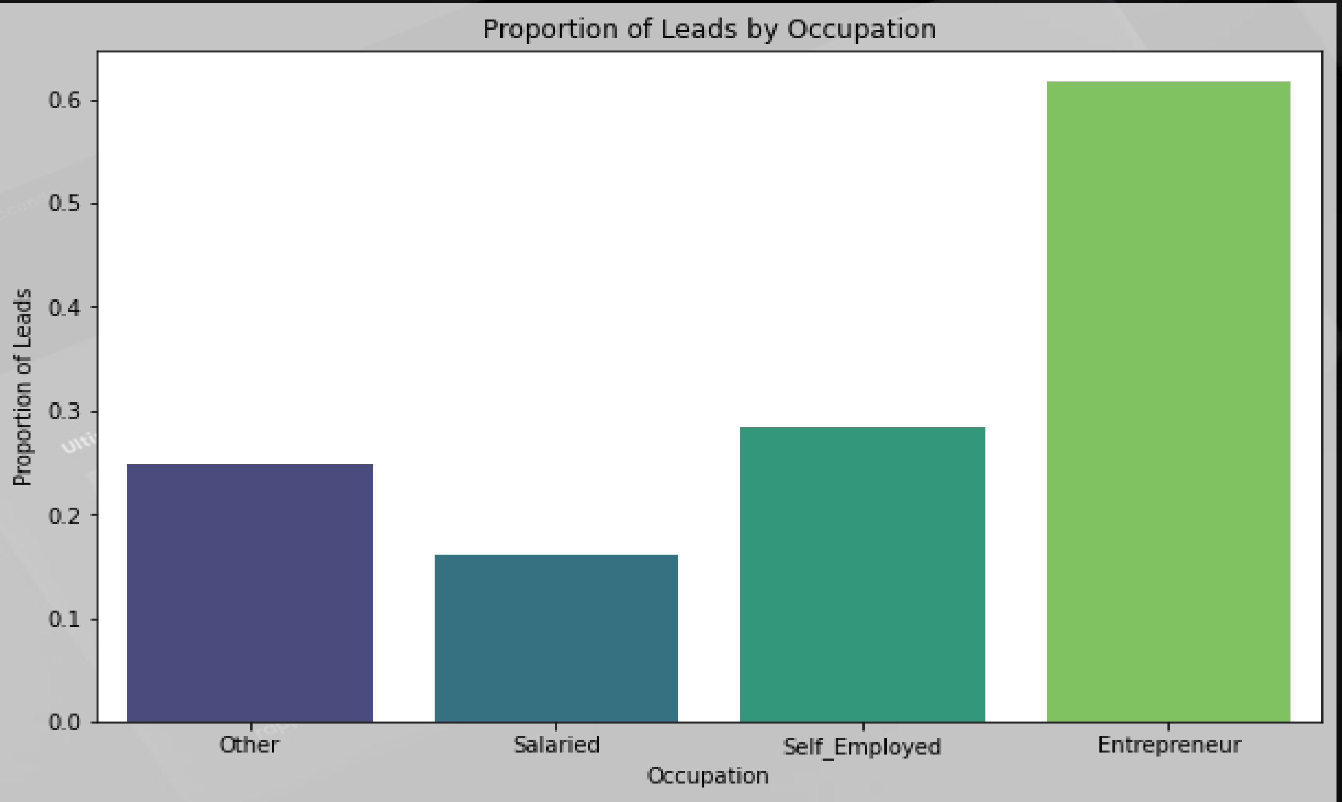


Entrepreneurs stand out as the occupation category with the highest proportion of leads (above 60%).

This suggests that entrepreneurs are highly interested in the financial product or service being offered.



Males seems to dominate the lead conversions



The pairplot reveals some separability between Is_Lead = 0 and Is_Lead = 1 based on features like Age and Avg_Account_Balance.

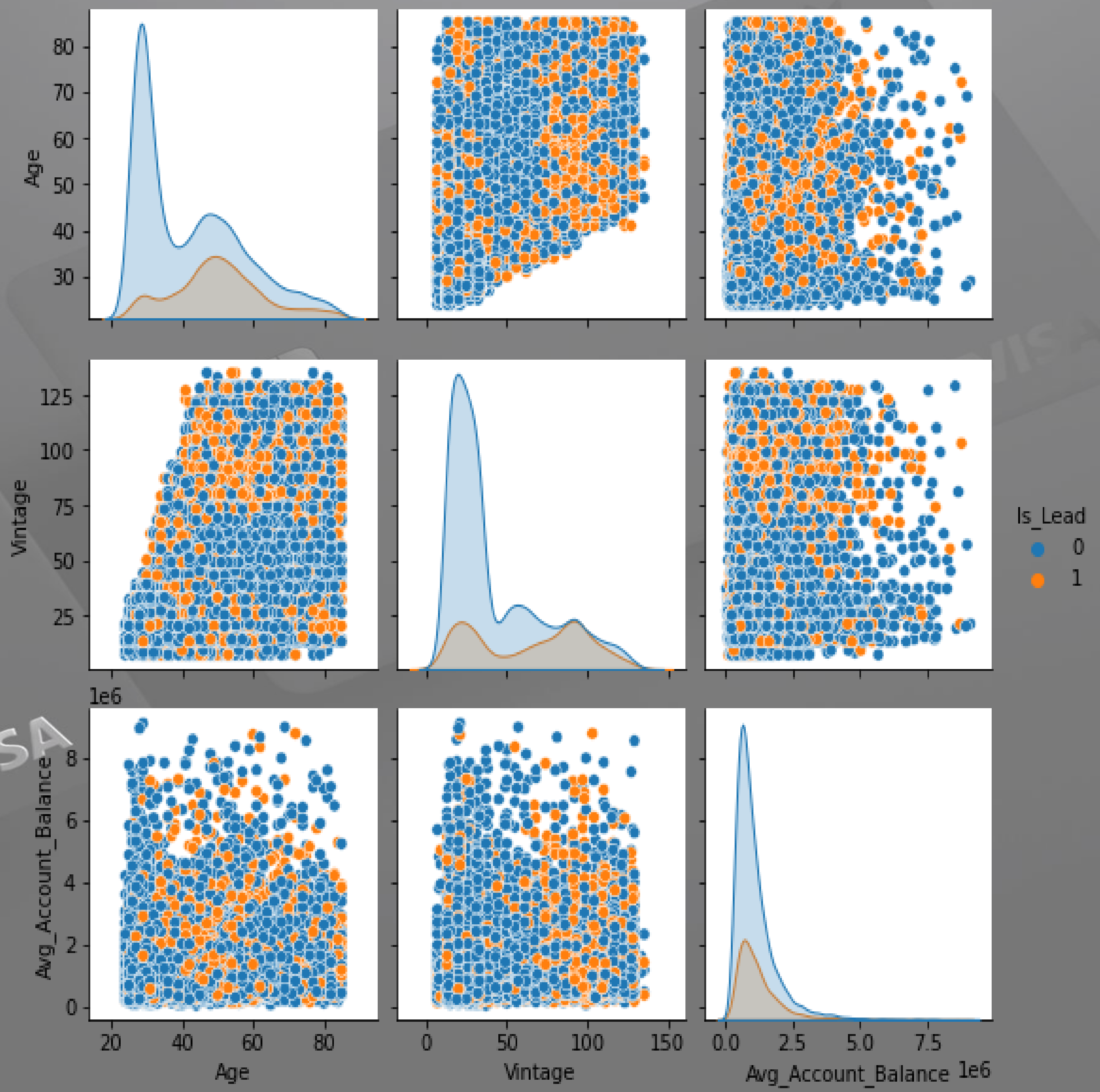
However, there's also overlap, indicating some noise in the data.

Imbalance in Target Variable:

The dataset is highly imbalanced, requiring resampling techniques or model adjustments

Metrics:

Use **Precision**, **Recall**, **F1-Score,** to evaluate the model, rather than accuracy, due to imbalance.



Feature Selection: The most Important Features are Occupation, Vintage, Credit_Product

Best Model Before Tuning:
Logistic Regression and **Gradient Boosting** had the highest precision (0.8496 and 0.8497).

Best Model After Tuning:
Decision Tree Classifier achieved the highest tuned precision (0.8567), followed closely by Random Forest Classifier (0.8550).

Thus, **Decision Tree Classifier** is the best model after hyperparameter tuning.

Model	Cross-Val Precision	Best Tuned Precision
Logistic Regression	0.8496 (Test)	0.8009
Random Forest Classifier	0.7429 (Test)	0.855
Gradient Boosting	0.8497 (Test)	0.8417
Decision Tree Classifier	0.7487 (Test)	0.8567

Evaluation of Models:

Logistic Regression (LR)

- **Precision**:
  - Train: **0.8496**, Test: **0.8496**
  - Consistently high precision, indicating the model has a strong ability to correctly identify positive cases when it predicts them.
- **Recall**:
  - Train: **0.4248**, Test: **0.4248**
  - Low recall, meaning the model misses a significant number of true positives.
- **F1-Score**:
  - Train: **0.5664**, Test: **0.5663**
  - Balanced but moderate F1-score, reflecting the trade-off between precision and recall.
- **Conclusion**: Logistic Regression offers high precision but sacrifices recall, which might not be ideal for imbalanced data where identifying true positives is critical.

Random Forest Classifier (RFC)

- **Precision**:
  - Train: **0.8350**, Test: **0.7435**
  - Train precision is good but overfits slightly, as the test precision drops.
- **Recall**:
  - Train: **0.4997**, Test: **0.4464**
  - Higher recall compared to LR, but still not ideal. Overfitting is noticeable.
- **F1-Score**:
  - Train: **0.6225**, Test: **0.5544**
  - Higher than LR on training but drops significantly on testing, again indicating overfitting.
- **Conclusion**: Random Forest improves recall slightly over LR but suffers from overfitting, as evidenced by the precision and F1-score drop on the test set.

Evaluation of Models:

Gradient Boosting (GB)

- **Precision**:
  - Train: **0.8500**, Test: **0.8497**
  - Very consistent between train and test, with minimal overfitting.

- **Recall**:
  - Train: **0.4252**, Test: **0.4251**
  - Similar recall performance to LR, which is relatively low.

- **F1-Score**:
  - Train: **0.5669**, Test: **0.5665**
  - Matches LR almost identically but offers slightly better stability.

- **Conclusion**: Gradient Boosting performs similarly to Logistic Regression but with slightly better generalization. Its precision-recall tradeoff is still an issue for imbalanced data.

Decision Tree Classifier (DTC)

Precision:

Train: **0.8628**, Test: **0.7490**

High train precision but overfits significantly, as test precision is much lower.

Recall:

Train: **0.4748**, Test: **0.4179**

Recall drops significantly on the test set, suggesting overfitting.

F1-Score:

Train: **0.6118**, Test: **0.5351**

Higher than LR and GB on training but lower on testing due to overfitting.

Conclusion: Decision Tree overfits the training data and generalizes poorly compared to other models.

Recommendation:

For this imbalanced dataset, the goal is to achieve a balance between **precision and recall** while minimizing overfitting. Based on the metrics:

Decision Tree Classifier:

- **Best Model After Tuning:** achieved the highest tuned precision (0.8567), followed closely by Random Forest Classifier (0.8550).
- Highest precision (critical for minimizing false leads).
- Balanced trade-off between precision and recall

Gradient Boosting:

- Offers the best balance between train and test scores, with minimal overfitting.
- Precision and recall are consistent, making it more reliable.
- Works well with imbalanced data and supports further tuning (e.g., adding class weights or resampling).

Random Forest:

1. Slightly better recall than GB, but suffers from overfitting.
2. If recall is a higher priority than precision, RFC could be improved with techniques like hyperparameter tuning or balanced class weights.