Jasper -> Jean-Paul

**Feedback on the Naive Bayes model:**

While SMOTE has been used. It might give a better result by undersampling the majority class.

The data has been preprocessed well. With data being dropped what is irrelevant and can only be seen as clutter.

The use of a confusion matrix is very good, this helps visualize how well the model is performing. I would advice using a bigger visual to make it look even better.

**Feedback on the Logistics Regression Model:**

The accuracy of this model is 86% which is 10% higher than the Naive Bayes model. With it being extremely good at predicting if a customer will stay by (0.95) 95% accuracy.

However the model is not good at predicting whether or not a customer is going to leave or close their accounts. This can be a result of a big majority of the customers used in this training set are likely to stay.

A good use of evaluation metrics have been used. Accuracy, precision recal, f1-score and a confusion has been used to test how accurate the model is.

**Feedback on the K Nearest Neighbour model:**

The KNN model shows a balanced performance with bot classes having a similar accuracy. This means that the model is quite consistent with predicting.

While the model is consistent with its predictions. The B2B prediction with a score of 71% accuracy is low. Non-B2B is even lower with 68% .

Something to consider is applying visuals to the text. While the text below the model explains what it does and what the results are, a visual might help if the reader wants to glance over it quickly.

Jean-Paul -> Jasper

**Feedback on the Naive Bayes Model**

The Naive Bayes model achieved an **88% accuracy** in predicting customer attrition, which is promising, especially in identifying customers who are likely to stay (precision: 0.92). However, its ability to identify customers who are likely to leave is weaker (precision: 0.65). This might be due to class imbalance, as most customers in the training set probably stayed with the company.

* **Class Imbalance**: One way to improve this imbalance is to try **undersampling the majority class** or applying **SMOTE (Synthetic Minority Over-sampling Technique)** to ensure more balanced representation. This could improve the model's sensitivity to customers likely to leave.
* **Confusion Matrix**: The confusion matrix effectively illustrates the model's performance, showing true positives and negatives clearly. Consider **increasing the visual size** of the confusion matrix so it’s easier to interpret the results at a glance.
* **SettingWithCopyWarning**: The code produces a SettingWithCopyWarning, which can be avoided by creating a separate copy of the DataFrame slice before applying the LabelEncoder. This small adjustment will make the code run without warnings.

**Suggested Next Steps**

Since the **precision for identifying customers likely to leave is lower**, I’d suggest trying **Logistic Regression** or **K-Nearest Neighbors (KNN)** to compare results. Logistic Regression can sometimes handle imbalances better than Naive Bayes, and KNN might offer a balanced performance.

If you proceed with **Logistic Regression**, here’s some feedback on typical results you might get:

* **Logistic Regression Performance**: Usually, Logistic Regression provides better predictions for binary classifications like customer attrition. If the model achieves around 86% accuracy, it would mean an increase over Naive Bayes, especially in predicting customer stay rates accurately. However, identifying customers likely to leave could still remain a challenge.
* **Evaluation Metrics**: Continue using precision, recall, f1-score, and a confusion matrix for consistency in comparing models.