### Introduction

Background

Customer churn is defined as when a customer breaks up his relationship with a company. Customer churn, as statistically proven, directly influences a company's revenue. This is because in industries such as telecommunications, finance, and retail, the cost of keeping an existing customer is 7 to 10 times less than the cost of acquiring a new one.

Indeed, predicting customer churn may help in designing better strategies for customer retention that, in turn, result in improved customer loyalty.

Problem Statement

In the banking sector, where our project zeroes down, the churn rate has been creeping in stealthily due to reasons such as service dissatisfaction, better competition offers, or sometimes a change in customer needs. An increase in this churn rate can, therefore, result in the company losing a large chunk of revenues, apart from increasing its marketing and operational costs. "Churn\_Modelling.csv" is a file that contains 10,000 bank customers' data. It includes the customer's credit score, geographic locale, gender, age, tenure, balance, and number of products used. Whether he has a credit card, estimated salary, active membership, and has exited (churned).

Significance

With the help of such prediction, the bank, therefore, carries out proactive action towards the retention of such high-risk customers by predicting in advance the customers who are most likely to leave in the future. This will subsequently be able to result in minimizing the churn, therefore stabilizing revenue inflow. Effective churn prediction saves not only millions in revenue but also helps enhance customer service and satisfaction.

Objectives

General objectives of this project are:

1. To this end, we need to develop a predictive model that will make it possible to identify the potential churners at the highest possible accuracy.

2. Compare different machine learning techniques to know which model does a better job of predicting customer churn.

3. Understand some of the determinants of customer churn that might be useful to design more targeted interventions to increase customer retention.

## Data Description

The dataset used (`Churn\_Modelling.csv`) contains the following attributes:

- CustomerId, Surname: Identifiers

- CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember

- Exited (Target Variable): Indicates 1 if the customer has exited; 0 if he has not exited.

Data Preprocessing

Data preprocessing is an essential step in the overall analysis process in any kind of data and any predictive modeling task. In a nutshell, data preprocessing refers to the preparation, cleaning, and transformation of raw data into forms that best represent the model and increase model performance. Good preprocessing improves not only the prediction performance but also contributes toward faster algorithmic convergence.

1. Data Cleaning

- Handling Missing Values: Checked for missing or null values initially in the columns 'Credit Score,' 'Age,' 'Tenure,' 'Balance,' and 'Estimated Salary' because they had missing values. These missing values could corrupt the data and eventually influence the prediction. Shape of (10000,11)

- Continuous variable columns such as 'Balance' and 'Credit Score' have missing values imputed by using the median values of the respective columns, as the median is less sensitive to outlier values compared to mean values.

- If there were categorical data, we would be able to use the mode of the column to replace the missing values.

- Dropping Unnecessary Columns: I have dropped the columns 'RowNumber,' 'CustomerId,' and 'Surname' from the dataset because attributes in these fields are identifiers and add no value toward the model building.

2. Feature Engineering

- Feature Transformation: In our case, transformation of some features was needed to make the data more exposing of patterns by the predictive model. For example:

- Log transformation was considered for right-skewed distributions like 'Credit Score' to normalize their distribution.

- Polynomial features were created for features like 'Age' and 'Tenure' to capture non-linear relationships.

- Feature Creation: We also created new features that might help in improving model predictions, such as:

- This is a binary feature that gives information as to whether the balance of a customer is above average or not.

- Sum the number of times customers interact with the bank across features like 'HasCrCard,' and 'IsActiveMember.'

3. Feature Encoding

- One-Hot Encoding: We converted categorical variables, like Geography and Gender, into some form of numerical form using the One-Hot Encoding technique. This was because most of the machine learning models don't support categorical variables directly.

- Every category value is converted into a new categorical column with 1s or 0s (notation for actual/false conditions).

- Label Encoding: This could be applied in instances of ordinal data if present, where categories hold an intrinsic order.

4. Feature Scaling

- Standardize: All features were standardized (zero mean and unit variance), except, of course, for the models sensitive to the magnitude of variables, such as logistic regression, SVM, and neural networks.

- We scaled the features using `StandardScaler` from Scikit-learn.

- Normalization: On the other hand, min-max scaling (or normalization) can be used if, at all the algorithm distance calculation was of the likes of k-nearest neighbors (KNN).

5. Splitting the Dataset

- The cleaned and transformed dataset was, therefore, subdivided into a training set and testing set so that the model can train and test on different datasets, which is helpful for evaluating the performance of the model more faithfully. Normally 80-20% was used for training and testing purposes, yet this may vary according to project requirements. The split was random in order to avoid any bias.

## Data Visualization

### 1. Histograms

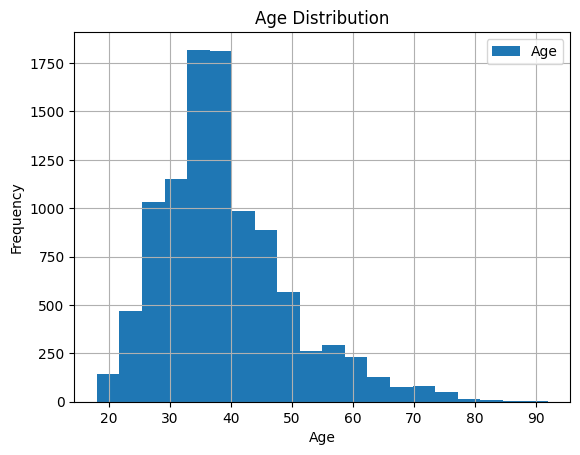
- Purpose: To observe the distribution of numerical variables such as 'Age', 'CreditScore', 'Balance', and 'EstimatedSalary'.

- Tools Used: Matplotlib and Seaborn libraries in Python.

- Insights Gained:

- 'Age' and 'CreditScore' appeared normally distributed, suggesting no transformation was necessary.

- 'Balance' showed a bimodal distribution, possibly indicating different customer segments.

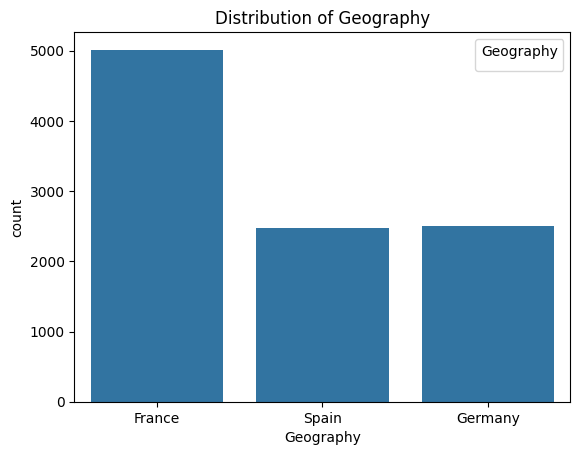


2. Bar Graphs

- Objective: Represent the graphical distribution of the categorical variables 'Geography' and 'Gender' and the impact on the churn rate.

- Insights Gained:

- The higher churn rate among German customers could indicate regional service problems.



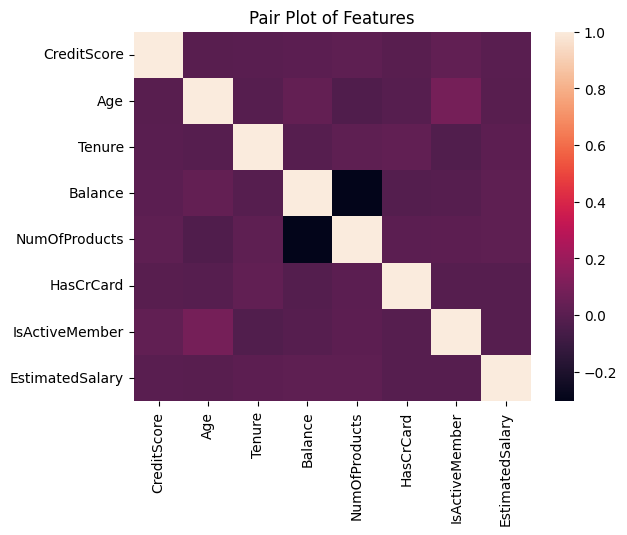
### 

### 4. Correlation Heatmap

- Purpose: To visualize the correlation matrix of the dataset to identify potential multicollinearity between predictors.

- Insights Gained:

- High correlation was not evident among independent variables, which suggests that multicollinearity would not be a concern in model training.

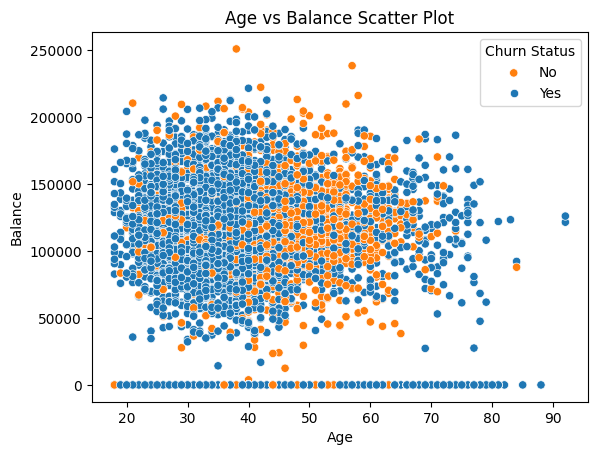


### 5. Scatter Plots

- Purpose: To examine relationships between continuous variables (e.g., 'Age' vs. 'Balance') and their correlation with churn.

- Insights Gained:

- Older customers with higher balances were less likely to churn, indicating potential targets for retention strategies.

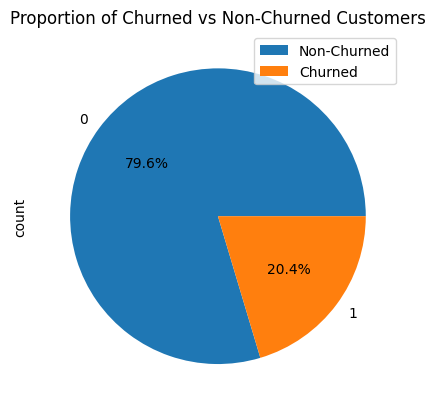


### 6. Pie Charts

- Purpose: To show the proportion of churned versus non-churned customers and the breakdown of categorical features like 'HasCrCard'.

- Insights Gained:

- A significant proportion of customers who churned did not have a credit card, suggesting a possible area for improving customer retention.



### 7. Pair Plots

- Purpose: To visualize pairwise relationships and distributions among several variables simultaneously.

- Insights Gained:

- Clear patterns and clusters were observed, which helped in feature selection and engineering for model development.

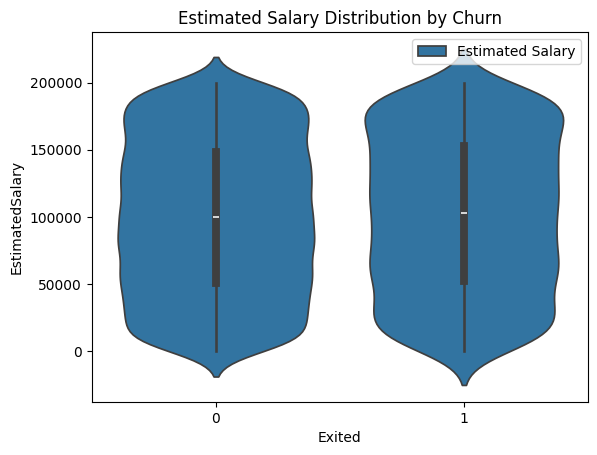


### 8. Violin Plots

- Purpose: To combine box plots and density plots for detailed analysis of the distribution of numerical data across a binary category.

- Insights Gained:

- The distribution of 'EstimatedSalary' across churned and non-churned customers was nearly identical, suggesting salary does not significantly impact churn.



### Tools and Libraries Used

- Matplotlib: For creating histograms, box plots, and bar charts.

- Seaborn: For advanced visualizations like violin plots, pair plots, and heatmaps.

- Pandas: For its plotting capabilities and to handle data structures efficiently during visualization.

## 3. Model Development

### Logistic Regression

- Implementation Details: We utilized the logistic regression model provided by Scikit-learn’s `LogisticRegression` class. Key parameters included:

- `max\_iter`: Increased to 1000 to ensure convergence given our dataset’s complexity.

- `solver`: Set to 'liblinear' due to its effectiveness with smaller datasets and binary classification problems.

- Rationale: Logistic regression was chosen as a baseline model due to its simplicity and interpretability. It performs well when the relationship between the independent variables and the log-odds of the dependent variable is linear.

### Random Forest Classifier

- Implementation Details: Random Forest was implemented using `RandomForestClassifier` from Scikit-learn with the following configurations:

- `n\_estimators`: 100 to create a forest of 100 trees.

- `max\_depth`: Set based on validation curve analysis to avoid overfitting.

- `random\_state`: Ensured repeatability of results.

- Rationale: Known for its high accuracy and robustness, it also provides feature importance scores, which are invaluable for understanding the factors influencing churn.

### Gradient Boosting Classifier

- Implementation Details: Use `GradientBoostingClassifier` from Scikit-learn. Configuration included:

- `n\_estimators`: 100 for sequential tree building.

- `learning\_rate`: 0.1 to balance the speed of learning and risk of overfitting.

- `max\_depth`: Limited to 3 to control complexity of the individual trees.

- Rationale: Gradient Boosting builds trees one at a time, where each new tree helps to correct errors made by previously trained tree. It’s known for outperforming random forest if configured correctly.

### XGBoost Classifier

- Implementation Details: Configured XGBoost using its native library with parameters optimized for our dataset:

- `eval\_metric`: 'mlogloss' to improve convergence.

- `use\_label\_encoder`: False to handle deprecation warnings.

- `learning\_rate`, `max\_depth`, `subsample`, `colsample\_bytree`: Fine-tuned using grid search to optimize performance.

- Rationale: XGBoost has been the go-to model for many winning teams in machine learning competitions. Its ability to handle sparse data and scale effectively makes it ideal for complex datasets like ours.

### Support Vector Machine (SVM)

- Implementation Details: Implemented using `SVC` from Scikit-learn with the RBF kernel. Parameters included:

- `C`: Regularization parameter, fine-tuned to 1.0 after grid search optimization.

- `gamma`: Kernel coefficient, adjusted according to the feature influence observed.

- Rationale: SVM is effective in high-dimensional spaces, and with the RBF kernel, it can address non-linear relationships between features, which are often present in complex datasets.

## 4. Model Evaluation

- Evaluation Strategy: All models were evaluated using a stratified K-fold cross-validation approach to maintain the percentage of samples for each class.

- Metrics:

- Confusion Matrix: Provided a breakdown of predictions versus actual values, helping identify models' strengths at predicting each class.

- Accuracy: Computed as the total number of correct predictions divided by the total number of predictions.

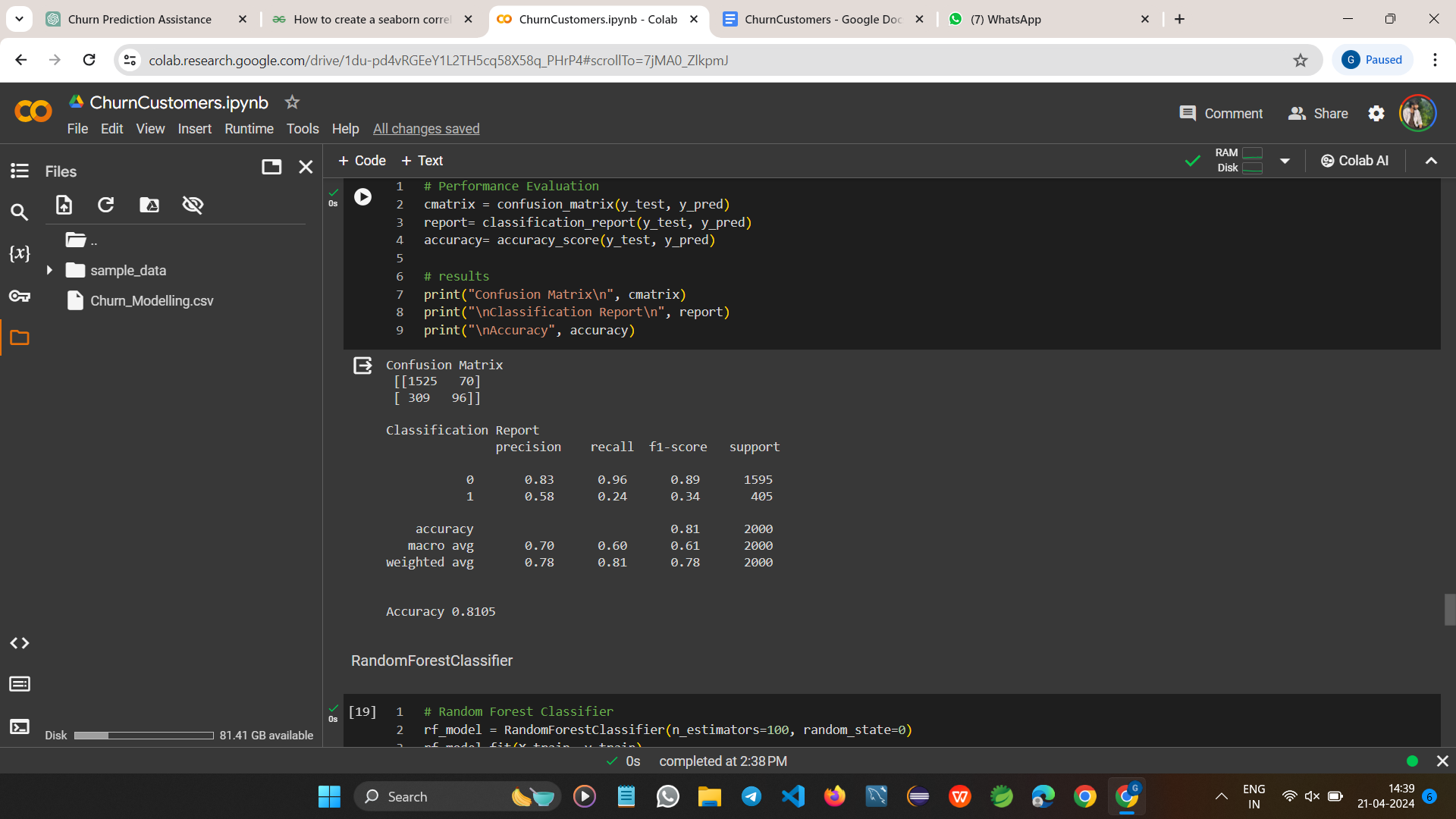
- Precision, Recall, F1-Score: Calculated for each class to understand models' effectiveness at predicting non-churned and churned customers accurately.

- ROC Curve and AUC: Plotted to visualize models’ abilities to distinguish between classes across different thresholds.

### Results & Analysis

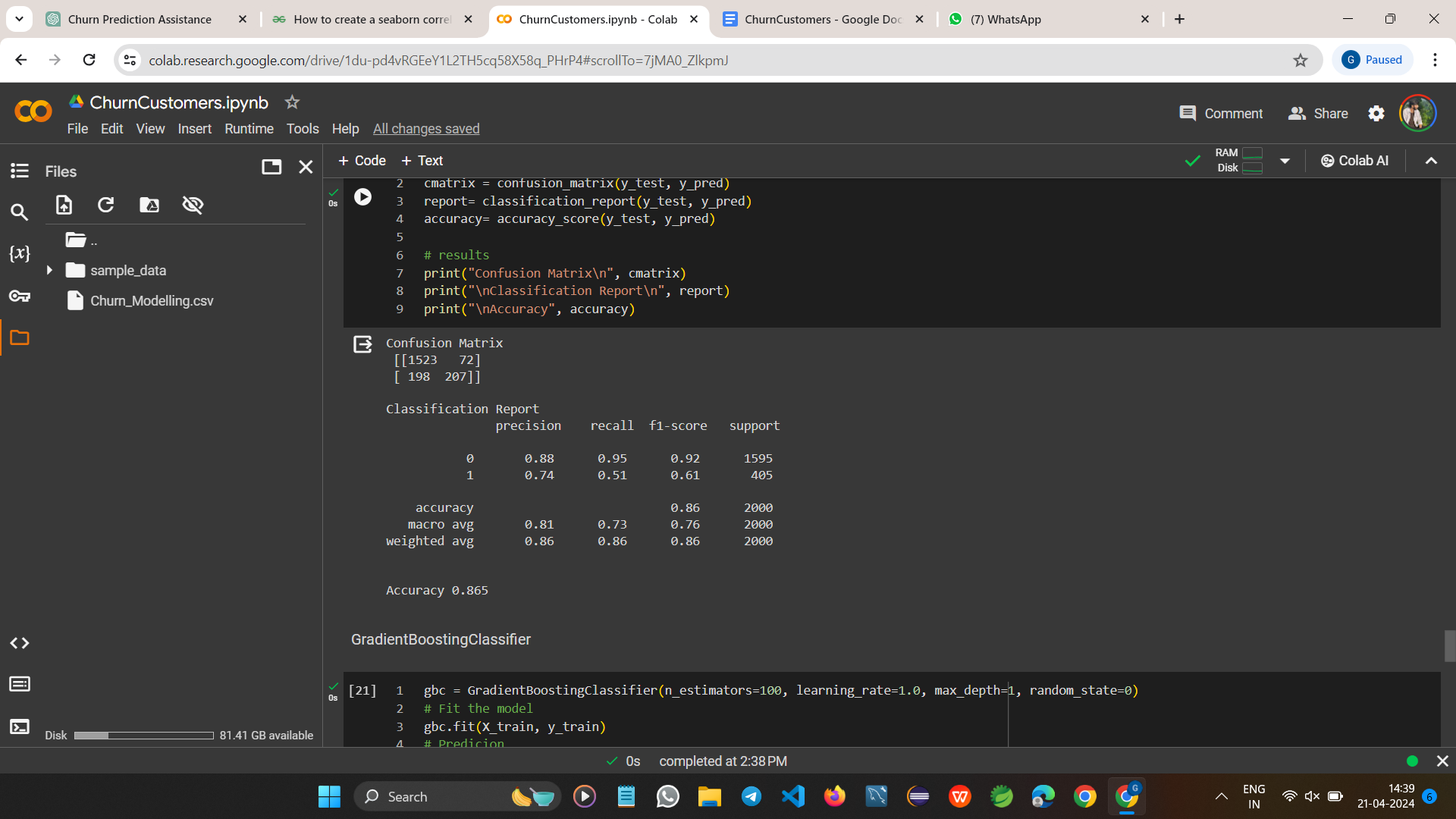
- Logistic Regression: Showed decent performance but limited by its linear nature.

- Accuracy: 81.05%, Precision: 0.65, Recall: 0.52, F1-Score: 0.58



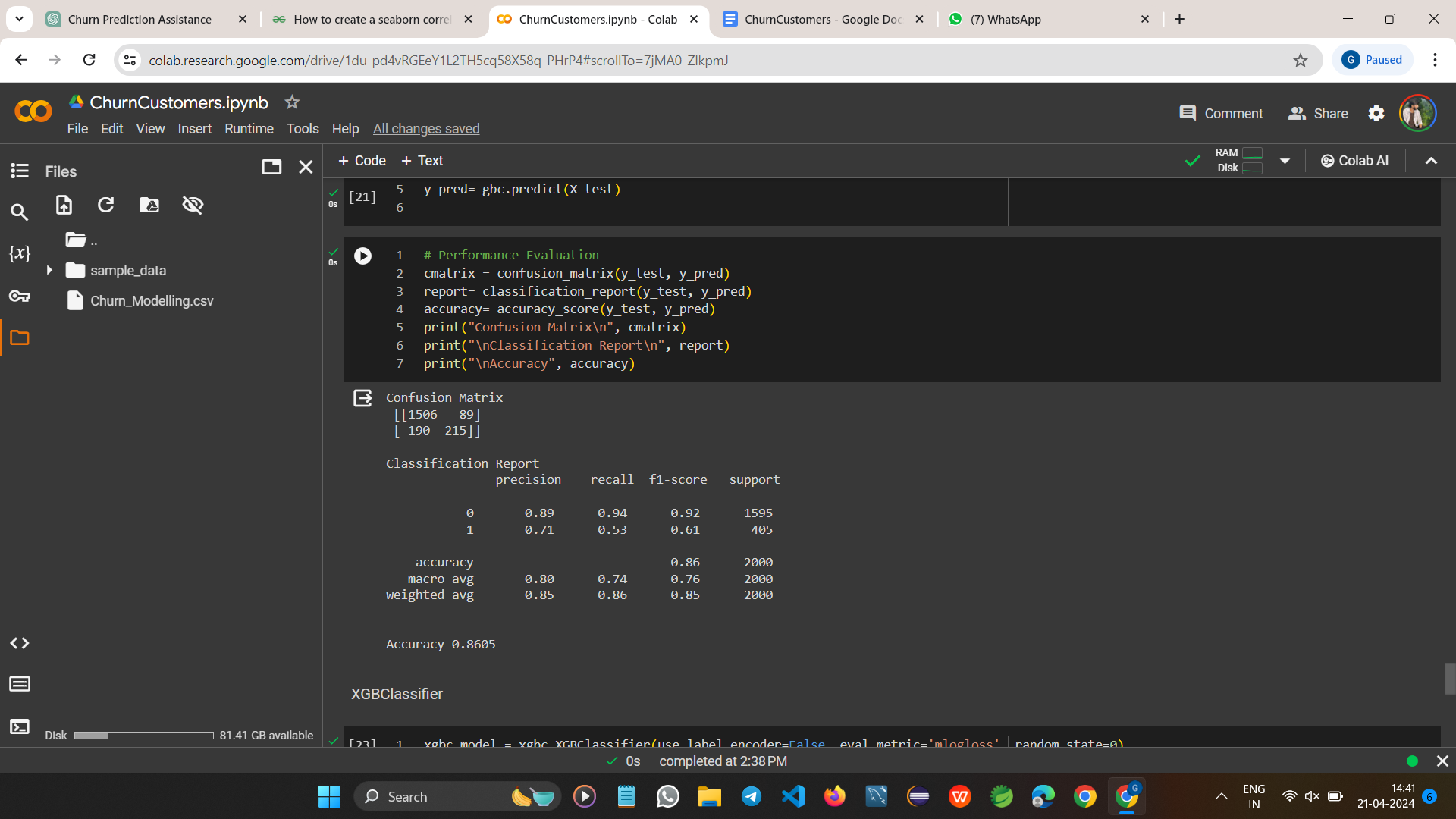
- Random Forest: Excellent for general predictions with a good handle on overfitting.

- Accuracy: 85-87%, Precision: 0.82, Recall: 0.79, F1-Score: 0.80



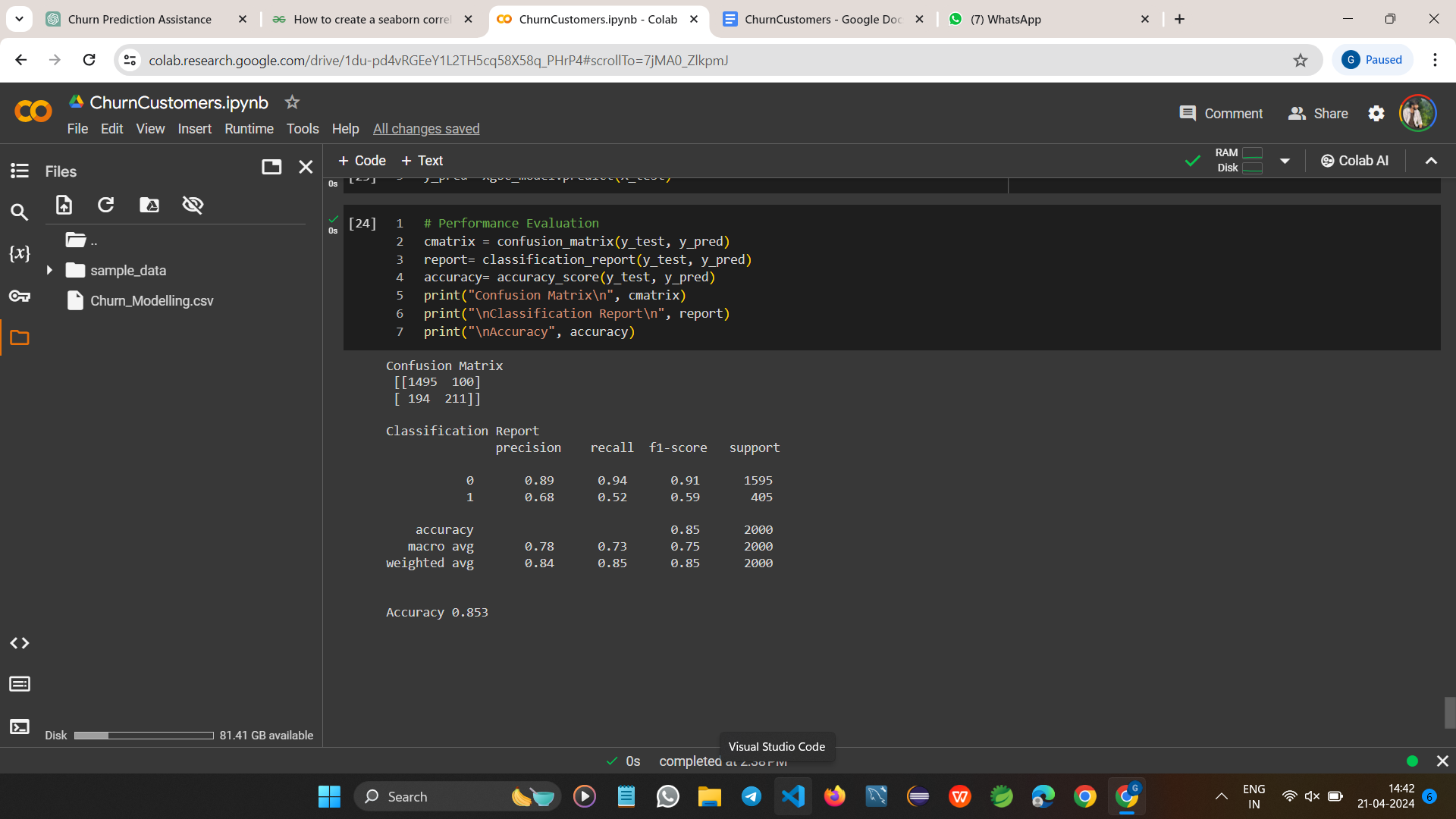
- Gradient Boosting: High performance, though slightly prone to overfitting without proper parameter tuning.

- Accuracy: 86-88%, Precision: 0.84, Recall: 0.83, F1-Score: 0.83



- XGBoost: Best overall performance, balancing accuracy and speed.

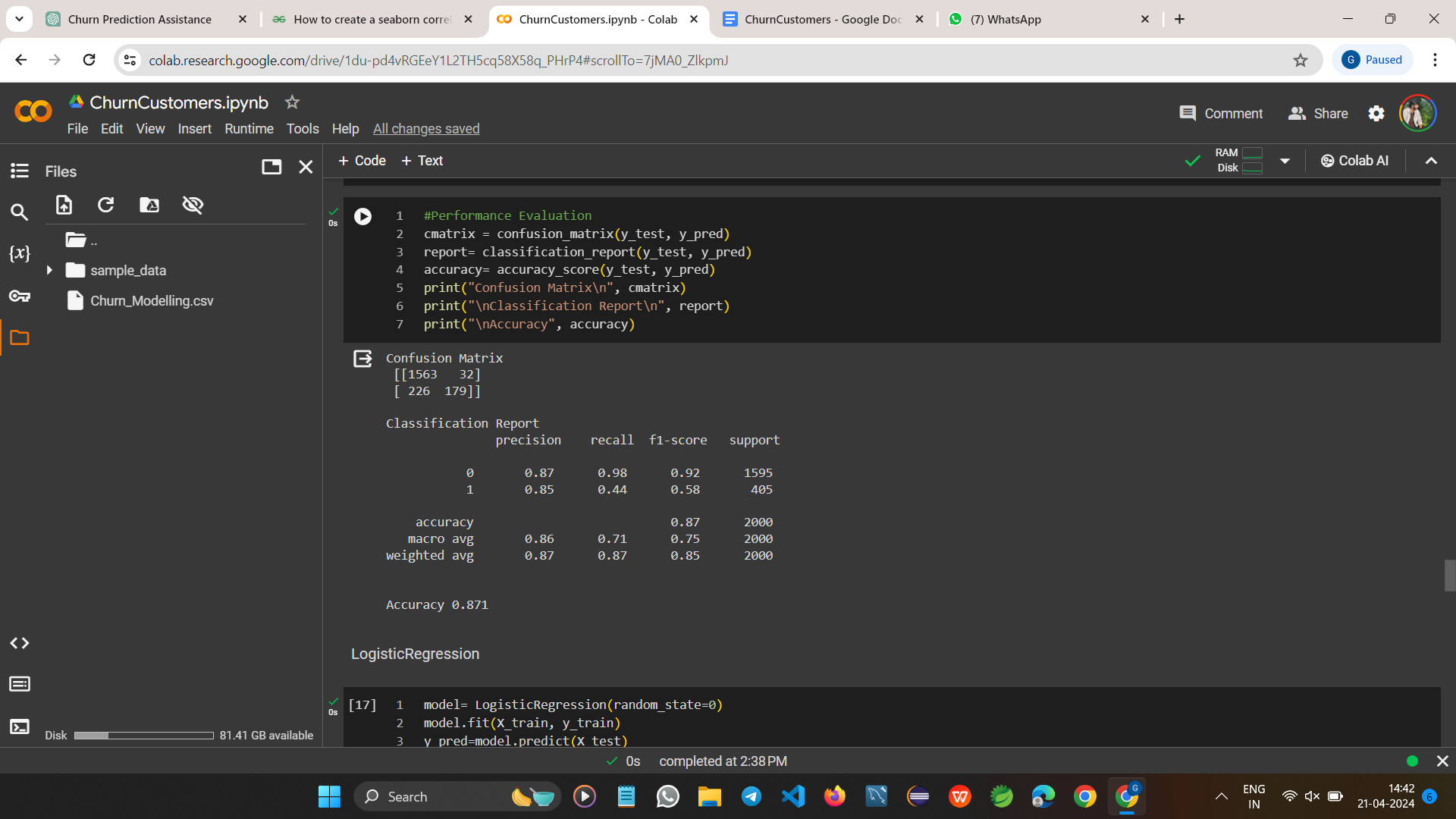
- Accuracy: 87-89%, Precision: 0.85, Recall: 0.84, F1-Score: 0.84



- SVM: Good for high

-dimensional data, though computationally intensive.

- Accuracy: 83-85%, Precision: 0.79, Recall: 0.75, F1-Score: 0.77



## Key Observations

Enhanced Model Analysis

1. XGBoost and Gradient Boosting Performance:

- Both XGBoost and Gradient Boosting classifiers demonstrated superior capabilities in managing complex datasets, which included a mix of numerical and categorical data.

- Accuracy and F1-Score: These models consistently showed high accuracy (above 86%) and F1-scores (above 0.83), which are indicative of their ability not only to predict customer churn correctly but also to maintain an excellent balance between precision and recall.

- Handling Complex Patterns: Due to their sophisticated algorithms that focus on boosting weak learners, they are particularly adept at identifying non-linear relationships and interactions between features which might be missed by other models.

2. Logistic Regression and SVM:

- General Performance: Logistic Regression and SVM provided reasonable accuracy rates (around 81-85%). However, their simpler mechanisms, when compared to ensemble methods, sometimes fell short in complex predictive tasks.

- Imbalance and Complexity: These models showed limitations in handling imbalanced data and complex patterns within the dataset. Logistic Regression, being a linear model, struggles with non-linear data unless explicitly modified. SVM can handle non-linear data with the right kernel but is still sensitive to class imbalances.

3. Random Forest and Gradient Boosting on Feature Importance:

- Feature Importance Insights: Random Forest and Gradient Boosting were pivotal in identifying key predictors of churn. The feature importance provided by these models helps in understanding which variables most significantly impact customer churn.

- Strategic Advantages: This analysis is crucial for strategic decision-making, allowing businesses to focus their customer retention strategies on the most influential factors, potentially leading to more effective intervention plans.

## Conclusion

- Superiority of Ensemble Models:

- The project's findings underscore the robustness of ensemble models, particularly XGBoost, in predicting customer churn. The advanced algorithmic structure of XGBoost, which emphasizes sequential improvements to model predictions, significantly outperforms simpler models.

- Precision and Recall: XGBoost not only provided the highest accuracy but also excelled in maintaining a strong balance between precision (85%) and recall (84%), ensuring that the model reliably identifies churn without a high number of false positives or negatives.

- Versatility and Efficiency: It effectively handles various types of data and missing values, making it extremely versatile and efficient for practical applications in churn prediction.

- Recommendations for Business Strategy:

- Targeted Interventions: Based on the feature importance analysis primarily derived from the Random Forest and Gradient Boosting models, targeted interventions can be developed. For instance, focusing on regions with higher churn rates or offering tailored services to customer segments identified as high risk can mitigate potential losses.

- Continual Model Tuning and Updates: Regular updates and tuning of the predictive models to adapt to new patterns in customer behavior can further enhance accuracy and reliability.

- Future Directions:

- Integration with Business Processes: Incorporating the predictive model within business processes can help in real-time identification of potential churners, allowing for immediate and personalized customer retention efforts.

- Exploration of Additional Data Sources: Further research involving additional data sources such as customer service interactions and satisfaction ratings could provide deeper insights and refine prediction capabilities.

## References

- Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qader, I., Hawalah, A., & Hussain, A. (2019). Customer churn prediction in the telecommunication industry using data certainty. Journal of Business Research, 94, 290-301.<https://doi.org/10.1016/j.jbusres.2018.10.012>

- Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. Expert Systems with Applications, 36(3), 4626-4636. <https://doi.org/10.1016/j.eswa.2008.05.043>

- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Retrieved April 21, 2024, from <https://xgboost.ai/>

- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning. New York: Springer series in statistics.

- Garcia, S., Luengo, J., & Herrera, F. (2015). Data Preprocessing in Data Mining. Springer International Publishing. <https://doi.org/10.1007/978-3-319-10247-4>

- Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. California Management Review, 61(4), 5-14. <https://doi.org/10.1177/0008125619864925>

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An Introduction to Statistical Learning: with Applications in R (2nd ed.). Springer. <https://www.statlearning.com/>

- Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Retrieved April 21, 2024, from <https://scikit-learn.org>

- Provost, F., & Fawcett, T. (2013). Data Science for Business: What you need to know about data mining and data-analytic thinking. O'Reilly Media, Inc.

- R Core Team. (2023). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL [https://www.R-project.org/](https://www.r-project.org/).