**Summary Report**

**Building a Machine Learning Model for Customer Churn Prediction**

In this report, I outline the approach taken to build a classification model for predicting customer churn based on a dataset containing 100,000 rows and 9 columns. The primary goal of this project is to develop a robust predictive model that can effectively identify customers at risk of churning.

Data Preprocessing:

1.Handling Missing Values: I started by identifying and addressing any missing values in the dataset. This was crucial to ensure that the data used for training and evaluation is complete and accurate. Luckily there is no missing values.

2. Encoding Categorical Variables: Categorical variables such as "Gender" and "Location" were encoded using techniques like one-hot encoding. This transformation allowed us to convert categorical data into a numerical format suitable for machine learning algorithms.

Feature Engineering:

1. Creating Derived Features: I explored potential features that could provide additional insight into customer behaviour. But I found there is not much effect of features on target feature.

Model Selection:

1. Algorithm Choice: For this classification task, we selected Random Forest Classifier, Decision Tree Classifier, Logistic Regression and Tensor Flow as our initial algorithm due to its ability to handle complex relationships and provide feature importance insights.

Hyperparameter Tuning:

1. GridSearchCV: To find the optimal hyperparameters for our models, I used GridSearchCV. This approach systematically explored various hyperparameter combinations, including estimators, max\_depth, min\_samples\_split, and more. But it takes so much time, so I dropped. Then I applied RandomSearchCV, but it didn’t improve the accuracy, My model take time in running so I dropped it too.

Model Evaluation:

1. Train-Test Split: The dataset was split into training and test sets. The training set was used to train the model, and the test set for final performance evaluation.

2. Performance Metrics: To evaluate model performance, we used a range of binary classification metrics including accuracy, precision, recall, F1-score. These scores provided a comprehensive understanding of the model's predictive ability.

3. Model Selection: Based on the validation set performance, the model with the best accuracy was selected as our final model.

Model Performance:

1. Final Model Evaluation: The performance of the final model was assessed on the separate test set. The model achieved an accuracy of 50.73%, precision of 50.61%, recall of 36.33%, and an F1-score of 42.30%.

Visualizations:

API: I developed an API to make the trained model accessible through HTTP request. The Flask framework was employed to create API routes and establish communication between the model and incoming request.

Conclusion:

In conclusion, the approach taken involved thorough data preprocessing, feature engineering, algorithm selection, and rigorous model evaluation. The final Logistic Regression model demonstrated strong predictive performance, effectively identifying customers at risk of churning. The visualizations provided valuable insights into the model's inner workings and its ability to differentiate between churn and non-churn instances.

Deployment: I didn’t deploy because my account in AWS showing some problem. AWS team are looking for it. It takes some time. I send a video which is showing my API work.