This review is conducted for Han Zhang's Churn Prediction Capstone Project per Applied Analytics Class by Elif Cakir

1. Overall GitHub Structure

It's good to see that there is a clear objective for the project stated in the README.md file. The README.md provides a clear and concise overview of the project objective, methodology and evaluation metrics. It effectively communicates the importance of predicting customer churn in the banking industry and the significance of the project outcomes. I found the project's focus on predicting customer churn in the banking industry to be both interesting and significant. Here are some suggestions for improvement:

- Provide more context about the dataset used: It would be helpful to include some information about the source of the dataset and its characteristics, such as the number of observations and features. This would give readers a better understanding of the data being used and how it relates to the project objective.
- Include a section on the results and conclusions: It would be beneficial to include a summary of the key findings from the project and the conclusions drawn from the analysis. This would help readers understand the significance of the project outcomes and the implications for the banking industry.
- The .gitkeep file is not necessary if there are other files in the folder, so it can be removed to declutter the repository.

2. Code/Jupyter Notebook and Code Output

The code is well-organized, clear, and easy to understand. Excellent job! In each sub section I included some further feedback for improvement.

Exploratory Data Analysis

The exploratory data analysis (EDA) performed on the dataset is informative and well-presented. The head(), shape, info(), nunique(), and isnull().sum() methods give a good overview of the dataset's structure and characteristics. The data visualization is well-done, and it helps to highlight some interesting patterns and relationships in the data.

- Add comments to explain what each code block is doing.
- Instead of using bank.head() to check the data, consider using bank.sample(5) to randomly select 5 rows to ensure that the data is correctly loaded.
- Consider visualizing the distribution of each numerical feature using histograms or density plots to gain more insights.

- Consider using more descriptive variable names instead of abbreviations (NumOfProducts vs. no_prod).

Feature Engineering

Your feature engineering process is well-structured and organized. You have dropped the unnecessary features, converted categorical variables to numerical variables using label encoding and one-hot encoding, and split the data into training, validation, and testing sets.

- It would be useful to provide more context on the dataset and the problem you are trying to solve.
- You might consider scaling the numerical features in your dataset before feeding them to a machine learning algorithm.
- It is good practice to include comments in your code to explain each step. This can make your code more readable and help others understand your thought process.
- You might consider using stratified sampling when splitting your data into training, validation, and testing sets. This can help ensure that the proportion of classes is roughly the same in each set, which is important for imbalanced datasets.

Logistic Regression

- The code could benefit from better markdown comments to explain the purpose of each block of code.
- There are some typos in the comments and variable names. For example, "hyperparamter" should be "hyperparameter"
- The code could benefit from more documentation. For example, the purpose of the "GridSearchCV" function and the meaning of the hyperparameters are not explained in the comments.
- Consider using more informative variable names. For example, instead of "model1" and "model2," it could be "logistic_regression" and "logistic_regression_tuned," respectively.

Random Forest

- The code is written as a series of operations carried out on multiple models. It would be better to separate the code for each variation of Random Forest into different functions, and make the overall code more modular.
- The code lacks commenting, which makes it difficult to understand the steps taken by the code.

- The code splits the data into a training set and a validation set, but it does not seem to use the validation set for anything other than calculating performance metrics. It would be better to use the validation set to perform early stopping during the training of the models.
- It is good to see that hyperparameter tuning has been performed, but it would be beneficial to also plot the results of the grid search to visualize the effect of the hyperparameters on the model's performance.

XGBoost

- Add comments to explain the purpose of each code block and function.
- Instead of defining and fitting each model separately, you could create a function that takes the hyperparameters as inputs and returns the trained model, the predicted labels, and the performance metrics.
- You can use cross-validation to estimate the performance of each model more accurately and avoid overfitting.
- Display the performance metrics in a table or graph to compare the different models easily.

Model Evaluation and Comparison

The code looks good and follows best practices for model evaluation and comparison.

- The use of np.random.seed(42) ensures that the results are reproducible. This is good practice, especially when comparing models.
- The code selects the final model (in this case, model9) and evaluates its performance on the test dataset. The code calculates accuracy, precision, recall, and F1 score, which are common metrics used for classification problems. This is good practice for evaluating the performance of the model on unseen data.
- The code also calculates the performance metrics for the training and validation sets. This is important to see if the model is overfitting or underfitting the data.

3. Project Report

The report provides a clear and concise explanation of the project objective, methods, and findings. The report is well-organized and follows a logical flow, starting with an introduction and ending with a conclusion. The use of subheadings to break down each section of the report makes it easy to follow and understand.

The exploratory data analysis section provides a good overview of the dataset and identifies some key insights that help understand the factors contributing to customer churn. The feature engineering section also provides a clear explanation of the different techniques used and the rationale behind them.

The methods section describes the different models used, their strengths and weaknesses, and the hyperparameter tuning approach taken. The results section presents the findings of the project, highlighting the performance of each model and the most influential predictors.

Your report provides a comprehensive analysis of customer churn in the banking industry and provides valuable insights for banks to take proactive measures to retain customers and improve overall customer satisfaction. One criticism for the report is that some more detailed explanations of the methodology used in each model could be helpful for readers who may be less familiar with these techniques.

You did such a great job on this report and project! I really enjoyed reading your report and seeing the data and modelling you performed! The data you generated was fascinating!