Churn Prediction Report

05/04/2024

AI Project Management

Liyang.FANG

Zihang.WANG

Part 1

Strategic Objectives:

The strategic objective of the AI project at RetailGenius is using data and AI support to analyse the behaviour and habit of the user, therefore, to improve other systems such as personalization and recommendation system etc. Through this method to enhance the user experience and optimize operations. And to predict which users are high possibility to lose, by identifying these customers, RetailGenius can implement targeted retention strategies to prevent churn and enhance customer satisfaction.

Key Performance Indicators (KPIs):

Churn Rate: Measure the percentage of customers who will churn within a period.

Prediction Accuracy: Evaluate the percentage of the user who really churned in this period.

Retention Rate: Monitor the percentage of customers retained after implementing retention strategies in a period.

Customer Satisfaction: To assess the feedback and review from the users after the implementation of retention strategies.

AI Contribution to Improving Customer Retention:

Early Identification of Churn Risk: AI algorithms can analyze historical customer data and identify patterns indicative of potential churn. By predicting churn risk early, it can realize the intention of customer’s leaving before they decide to do it.

Personalized Retention Strategies: AI enables to segment customers based on churn risk and preferences. Personalized retention strategies, such as tailored promotions, loyalty rewards, or proactive customer support, can be use in each different customer specially because they all have their own segment.

Optimized Marketing Campaigns: AI-powered analytics can identify high-value customers and optimize marketing campaigns to focus on retaining these customers. Therefore, through this method it can maximize the impact of the resources and to concentrate on the high-value customer.

Project Design

Data:

Relevant Data Sources:

Customer Relevant Information: Age, gender, job, income etc.

Transaction History: Purchase history, average purchase value.

Engagement Metrics: Website/app interactions, time spent.

Customer Support Interactions: Queries, feedback.

Potential Challenges:

Data Quality: The data for the model may have the problem of inconsistent, incomplete, or inaccurate. That will affect the performance of the model.

Data Integration: The shape of the datasets may be different; It probably be hard to integrate all those datasets into the same shape. Sometimes may have to drop some.

Privacy and Compliance: To ensure the data obey the regulation privacy.

Dataset Imbalance: The dataset may have the problem of setting train and test split; it depends on the number of potential churned users. If the number is too small, it will cause the overfit problem of the model.

Models & Deployment:

AI Models:

Logistic Regression

Decision Trees

Random Forests

Gradient Boosting Machines

Model Training, Validation, and Testing:

Training: Train models on historical data, optimizing accuracy, precision and recall.

Validation: Validate model performance by using cross-validation techniques to ensure generalizability.

Testing: Use train_test_split and to evaluate final model performance with the test dataset to assess the performance in real-world.

Model Versioning and Serving:

Use version control systems (e.g., Git) for tracking model versions.

Deploy models using AWS service for easy deployment and scaling.

To test the model and API to make sure everything is working without bugs.

Monitoring:

Performance Monitoring:

Track model metrics (e.g., accuracy, precision, recall) over time to detect performance degradation.

Monitor data distribution shifts and concept drift to identify changes in the underlying data.

Implement automated alerts for anomalies or deviations from expected behaviour.

handling model drift and maintaining model accuracy:

Retrain models periodically using updated data to maintain accuracy.

Implement retraining triggers based on predefined thresholds for model performance degradation.

Continuously evaluate model performance against evolving business goals and objectives.

Project Team

Roles and Expertise:

Data Analyst: Pre-processing data and analyse data.

Data Scientist: Build predictive models and optimise it.

Data Engineer: Develop data pipelines, manage data infrastructure.

Machine Learning Engineer: Deploy and maintain machine learning models.

Business Analyst: Translate business requirements into technical solutions.

Project Manager: Coordinate project activities, manage timelines and resources.

Cross-Functional Collaboration:

Regular meetings and workshops involving all team members.

Use of collaborative tools like project management software and shared document repositories.

Encouraging open communication and feedback across all levels of the team.

Skills and Expertise:

Data Scientist: Strong statistical and analytical skills, proficiency in programming languages like Python or R, experience with machine learning algorithms.

Data Engineer: Expertise in data processing and manipulation, knowledge of database systems and cloud platforms.

Machine Learning Engineer: Experience in model deployment and productionization, familiarity with containerization and microservices architecture.

Business Analyst: Understanding of business objectives and KPIs, ability to translate business requirements into technical specifications.

Project Manager: Leadership and organizational skills, ability to manage resources and mitigate risks.

Alignment with the Project Strategy:

Regular alignment meetings with key stakeholders to ensure project goals are understood and prioritized.

Clearly defined project objectives and success criteria communicated to the team.

Continuous monitoring of progress against project milestones and strategy.

Collaboration with other Departments:

Regular meetings and workshops involving stakeholders from marketing, customer support, and other relevant departments.

Sharing of insights and findings from the AI project to inform marketing strategies, customer engagement initiatives, and product development.

Project Governance & Communication

Stakeholders:

Executive Leadership: Provide strategic direction and resource allocation.

Business Stakeholders: Define business objectives and requirements.

Data Team: Execute AI project tasks and deliverables.

Technology Team: Provide infrastructure and technical support.

Governance Instances:

Regular status meetings and progress updates.

Steering Committee: High-level oversight and decision-making authority.

Change Control Board: Review and approve changes to project scope or requirements.

Communication Plan:

Regular reports and presentations to stakeholders.

Tailored communication channels for technical and non-technical audiences.

Documentation of model outputs and predictions for reference.

AI Project Management Methodology

Agile Methodology:

Suitable for iterative development and quick adaptation to changing requirements.

Facilitates cross-functional collaboration and stakeholder involvement throughout the project lifecycle.

Part 2

The Structure of the whole project (Data Handling & Model Saving)

Three joblibs for saving two transformers and one model in the machine flow. By saving these trained transformers and model, we could let the users implement the predictions. They can call the stored model directly. For the dataset, to save the dataset as xlsx format and let the user choose the feature they want to make the prediction. To achieve this function, there is a config module for saving all the parameters need of the model. Once the requirements of the model come, the parameters will change immediately in the config and be implemented in the model according to the new ones.

For the model part, there are two main parts to consist of it and one part to implement the whole model. We have preprocessing and train which are the dealing with data preprocessing for the preparation of the using by model and the training of the model, once it has been trained it will be saved into the joblib for users. After connecting the config, preprocessing and train, we made an inference script to implement the whole model rather than running each part one-by-one. Finally, return the prediction and the accuracy to the interface for the user to check whether they are satisfied with the result or change the features they want to optimise it.

Propose Solution of Cloud Deployment (Interface for User Interaction)

Data Storage: Use AWS S3 to store the data securely in the cloud.

Two choices for Hosting:

EC2: For full control over the hosting environment and to handle computing tasks.

Elastic Beanstalk: For easier setup and management of the web application.

Model Inference:

Use AWS Lambda for running the model without setting up servers. Lambda can also respond to user inputs.

Amazon API Gateway:

Set up an API to let users interact, sending their data and getting predictions back.

Configuration Management:

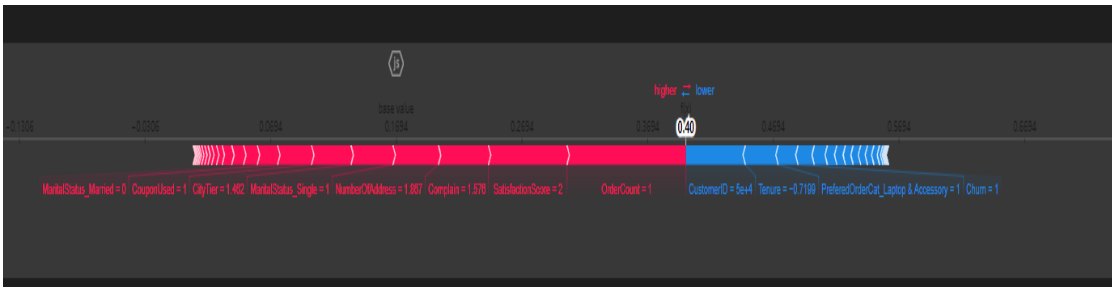
Use AWS RDS (for SQL databases) or DynamoDB (for NoSQL databases) to manage the application's settings and configurations.

Monitoring:

Amazon CloudWatch: To keep track of your application's health and usage.

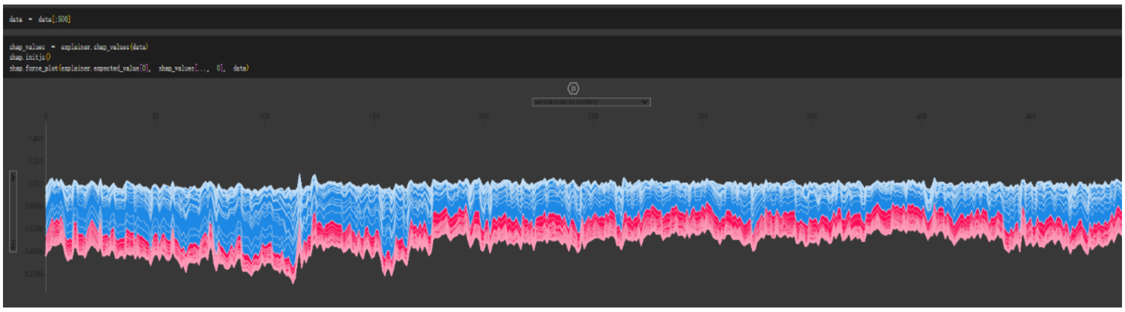
Part 3

Based on the experiment's results, we can analyse from the graph to get the which factor is the one that affect the result of the model mostly in positive and also what factors affect in negative way.



For example, based on the graph above, which is a Force plot. This graph means the horizontal axis represents the SHAP values of the features, which is a measure of the impact of each feature on the model's prediction. Higher values indicate a larger positive influence of the feature on the prediction result, while lower values indicate a larger negative influence. 0.4 is the baseline value (expected output) of the model, those values on the right side represents positive influences and the left side are negative influences.

Based on the graph we can see those features such as Customer ID, Tenure, PreferredOrderCat_laptop, Accessory, and Churn have SHAP values ranging from approximately 0.4 to 0.5694. This means they have much higher impact on the prediction result than the other features.

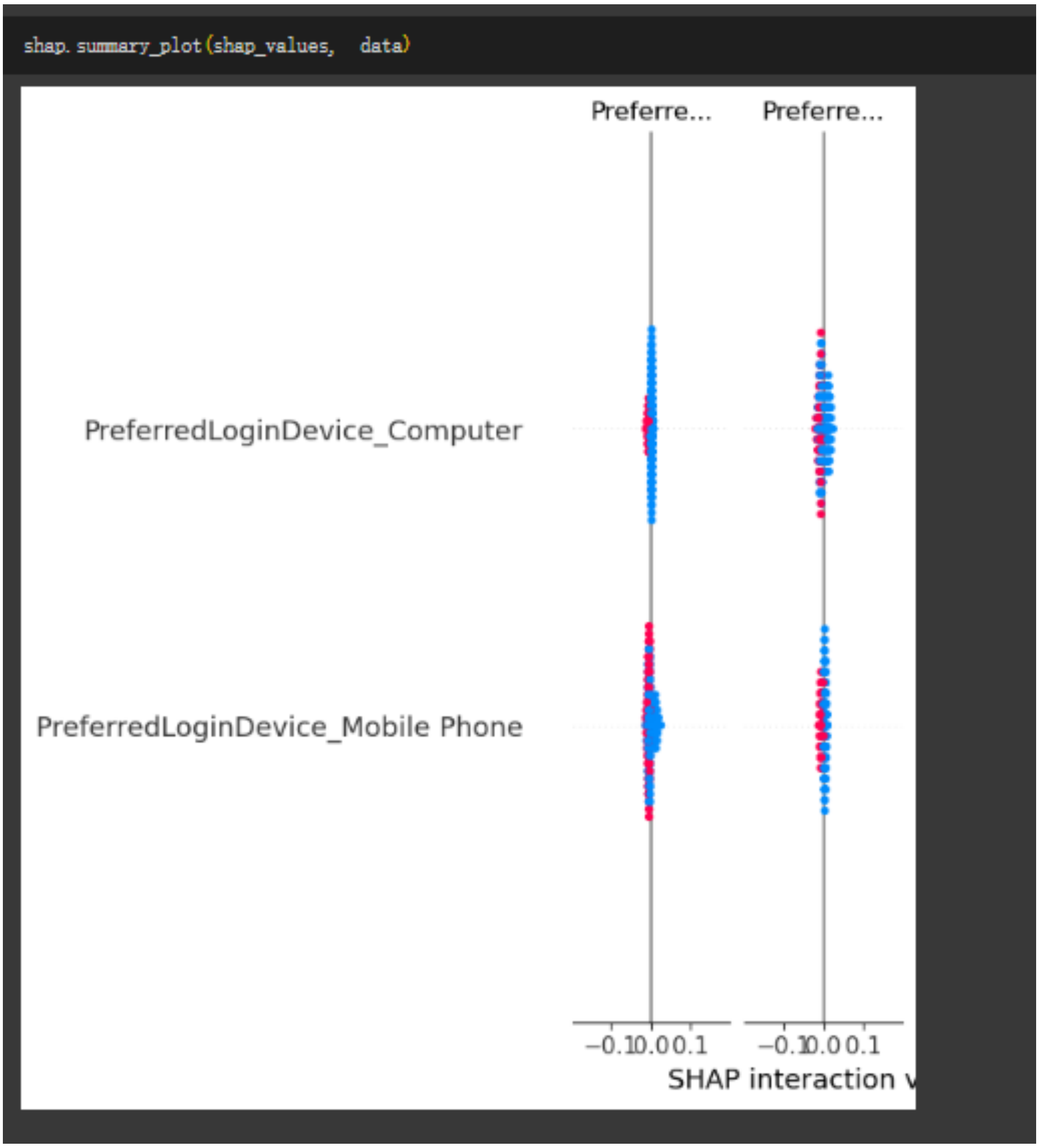


```
data = data[:500]
```

```
shap_values = explainer.shap_values(data)
shap.initjs()
shap.force_plot(explainer.expected_value[0], shap_values[..., 0], data)
```

The x-axis typically represents the SHAP value which is the impact in the output of the model. The y-axis shows the features for a particular instance. The red colour indicates features that push the prediction higher that is the positive impact, and the blue colour represents features that push the prediction lower which is the negative impact.

The difference between this map and the Force plot above is that we can select the feature from the dataset and to see the overall effect it has to the model based on the 500 elements from dataset. Finally, the width of the colour bands indicates the magnitude of the SHAP values for each feature; wider bands suggest a greater impact on the model output.



```
shap.summary_plot(shap_values, data)
```

Preferre...
Preferre...
PreferredLoginDevice_Computer
PreferredLoginDevice_Mobile Phone
−0.10.0 0.1
−0.10.0 0.1
SHAP interaction v

There are two features in this plot which are PreferredLoginDevice_Computer and PreferredLoginDevice_MobilePhone. Each dot on the plot represents a SHAP value for a single instance in the dataset. Same as above, the blue dot means the low prediction / negative impact and the red dot means the high prediction / positive impact.

The PreferredLoginDevice_Computer feature shows many blue dots on the left and a little red dot on the right. Indicating that for most of instances, having a computer as the preferred login device decreases the model's prediction, whereas for others, it increases the prediction. The same is true for PreferredLoginDevice_MobilePhone but with less pronounced effect and it shows more balanced.