ADR 1 Choosing the Machine Learning Model for Churn Prediction

The Netflix Churn Predictor project aims to forecast user churn on entertainment platform through machine learning algorithms. In the Netflix Churn Prediction project, the pivotal choice lies in selecting the most fitting machine learning model for accurately forecasting user churn.

Decision

Opted for the Gradient Boosting Machine (GBM) model to predict churn effectively.

Rationale

GBM stands out for its exceptional accuracy in prediction and adeptness in handling intricate feature relationships. While considering alternatives, found that Random Forest models, although robust, might not capture subtle patterns as effectively as GBM. Despite being simple and interpretable the logistic Regression may lack the necessary complexity for accurate churn prediction. Neural Networks, though promising in accuracy, raised concerns about overfitting and resource demands.

Status

Accepted

Consequences

Utilizing GBM ensures accurate predictions and adept handling of complex features. However, it might demand more computational resources during training compared to simpler models like Logistic Regression, and it may not exhibit the same strength as Random Forest models in certain scenarios.

ADR 2 Deployment Strategy for Netflix Churn Predictor

Context

Determining the deployment strategy for the Netflix Churn Predictor is paramount for seamless integration with existing systems and future scalability, and growth.

Decision

Opted for a microservices architecture deployed on a cloud-based platform.

Rationale

Microservices architecture offers the advantage of deploying and scaling different components independently, important for future growth. While monolithic deployment offers simplicity initially, it may hinder scalability as the application grows. Although Docker ensures portability through containerization, it introduces complexities in coordination. A serverless architecture, while offering automatic scalability, may lead to delays and lock-in concerns.

Status

Accepted

Consequences

Adopting microservices architecture enables independent deployment and scalability but requires meticulous design to manage communication and data consistency. While enhancing scalability and flexibility, it may introduce complexities in system management.

ADR 3 Selecting External Service for Data Storage

Context

Selecting the appropriate external service for data storage is critical to ensure scalability, reliability, and performance for the Netflix Churn Predictor.

Decision

Choose Amazon DynamoDB as the external service for storing user data and predictions.

Rationale

Amazon DynamoDB is selected for its seamless integration with the AWS ecosystem and scalability features. While Google BigQuery offers powerful analytics capabilities, it may have limitations in real-time data access. MongoDB Atlas provides flexibility but may require more management overhead.

Status

Accepted

Consequences

Utilizing Amazon DynamoDB ensures seamless integration with the AWS ecosystem and scalability features. However, it may incur higher costs for large datasets compared to alternatives like MongoDB Atlas.

ADR 4 Implementing Authentication and Authorization Mechanism

Context

Since the Netflix Churn Predictor handles sensitive data, it's crucial to prioritize its security. Unauthorized access could compromise data integrity and lead to serious breaches. Therefore, need a strong authentication and authorization mechanism to protect against such threats.

Decision

OAuth 2.0 for authentication, paired with role-based access control (RBAC) for authorization.

Rationale

OAuth 2.0 is chosen for its standard security features and widespread use across industries. While JSON Web Tokens (JWT) offer authentication without maintaining state, they lack centralized control, making OAuth 2.0 a better fit for our needs. Also, Basic Authentication, while simple, has security vulnerabilities that make it unsuitable for our purposes.

Status

Accepted

Consequences

Implementing OAuth 2.0 ensures secure authentication, strengthening our defenses against unauthorized access. However, it may add complexity to configuration, requiring careful management. RBAC enhances security with detailed access control policies, but it needs close oversight to manage and enforce effectively.

ADR 5 Selection of Real-time Notification Service

Context

Selecting a reliable real-time notification service is crucial for ensuring users stay informed about significant updates and alerts.

Decision

Amazon Simple Notification Service (SNS) for real-time notifications.

Rationale

Amazon SNS is chosen for its scalability, reliability, and seamless integration with the AWS ecosystem. While Firebase Cloud Messaging supports mobile platforms, it may have limitations for web-based notifications. Apache Kafka offers flexibility but may require more operational expertise.

Status

Accepted

Consequences

Using Amazon SNS ensures scalability and reliability, but may incur costs based on message volume. Integration with the AWS ecosystem provides seamless operation but requires adherence to AWS's service offerings and pricing.