Home Task - ML Engineer

Home Task: Customer Risk Segmentation and Off-boarding Identification

Objective

The goal of this task is to assess your ability to segment customers based on risk using labeled customer and transaction data, then apply your model to an unlabeled dataset to identify customers for off-boarding.

Additionally, you will describe:

- How to deploy this model on Google Cloud Platform (GCP).
- How to monitor its performance (which metrics will you look for, alerts, dashboards, KPIs, all? your choice.)
- How to make sure it can scale 10x (which of those metrics need attention and what do you suggest to improve the infra or the model when it is stressed due to traffic)

Task Description

You are provided with **two datasets**: one labeled for model training, and another unlabeled for prediction and inference.

Provided Data:

1. **Labeled Customer and Transaction Data** (for training your model):
    - **Customer Data - with ultimate account state**
        - SUSPENDED = TRUE POSITIVE FOR OFF_BOARDED DUE TO RISK
        - ACTIVE = GOOD USER WITHIN RISK APPETITE
    - **Transaction Data - transaction history of users from labeled cohort**

2. **Unlabeled Customer and Transaction Data** (for inference):
    - Customer Data
    - Transaction Data (same structure as the labeled dataset).

Part 1: Model Training for Customer Risk Segmentation

1. **Data Exploration & Preprocessing**:
    - Explore the labeled customer and transaction datasets to understand the data.
    - Handle missing values, outliers, or any anomalies in both datasets.
    - Perform feature engineering to create useful features for customer risk classification.
2. **Model Training**:
    - Using the labeled customer and transaction data, build a machine learning model to predict customer SUSPENDED outcome.
    - You may use classification algorithms such as Decision Trees, Random Forests, Gradient Boosting, or other approaches that you deem appropriate.
    - **Explain your choice of model** and provide justification for your approach, including how you tune hyperparameters (if applicable).
3. **Model Evaluation**:
    - Evaluate your model using appropriate metrics such as accuracy, F1-score, precision, recall, etc.
    - Perform cross-validation to ensure the model generalizes well to unseen data.
    - Include a discussion on the **feature importance** (i.e., which customer or transaction features were most influential in predicting the risk category).

Part 2: Applying the Model to Unlabeled Data & Identifying Off-boarding Customers

1. **Risk Prediction**:
    - Apply the trained model to the **unlabeled customer and transaction dataset** to predict SUSPENDED outcome
2. **Off-boarding Criteria**:
    - Define criteria for off-boarding customers based on your prediction and transaction patterns.
        - Clearly explain how these criteria align with business objectives.
3. **Implementation**:
    - Using the predicted risk categories and transaction data, generate a list of customers that should be off-boarded.
    - Justify why each of these customers meets your off-boarding criteria

---

Part 3: Reporting & Documentation

1. **Summary**:
    - Summarize your findings, including:
        - The performance of your model on the labeled dataset.
        - The characteristics of customers predicted to fall into each risk segment (Low, Medium, High) in the unlabeled data.
        - The final list of customers identified for off-boarding, along with reasons why they were selected.
2. **Code**:
    - Submit your code in a modular and well-structured format (Python scripts or Jupyter notebooks).
    - Ensure your code is well-documented with clear comments and explanations.
3. **Assumptions and Limitations**:
    - State any assumptions made during the analysis and modeling.
    - Discuss limitations in your approach or the provided data.

Part 4: Deploying the Model on Google Cloud Platform (GCP)

Please describe the following steps in order to deploy the model into a production environment. The more detail, the better :)

1. Containerization and Packaging

Before deploying the model, containerize it for easy deployment and scaling. You can use Docker to package the model along with its dependencies.

- Docker Setup: Write a Dockerfile that includes all necessary dependencies (e.g., Python, ML libraries such as Scikit-learn or TensorFlow).
- Best Practice: Ensure that the Docker image is lightweight and optimized for faster loading.

2. Deploying with GCP Services

Assume you deploy the model using Google Kubernetes Engine (GKE). How would you do that?

Also: Endpoint Creation: Describe a RESTful API that allows the model to be accessed via a REST API, enabling real-time or batch inference from other microservices.
You can use
https://prod.deblock.com/ai/ as the example root url for your APIs.

3. Data Input and Output Handling

Where would you store the inputs/outputs within GCP? Describe the options and why.

4. Monitoring and Logging

How would you collect logs from the model (errors, predictions, etc.) in real-time?

How would you create custom dashboards and alerts to monitor model performance (latency, prediction accuracy, request volume). Please define Key Metrics to Monitor

5. Scaling the Model

To ensure the system can scale 10x, focus on these strategies:

- Auto-scaling: How would you achieve this using GCP?
- Model Optimization: How would you achieve this?

What other techniques would you consider? Add a Pros/Cons table.

6. CI/CD Pipelines

Describe your ideal pipeline to go from scratch into production.