Senior ML Engineer Take-Home Technical Assessment

Overview

Welcome to the Senior ML Engineer take-home technical assessment. You will be working with a sample data set from one of our affiliate partners. This challenge aims to evaluate your ability to explore, analyze, and build production level models. We are looking for an understanding of the assignment, insights into your thought process and approach, answers to the questions, a clean presentation of your work, and a correct use of basic ML Engineering tooling. We've included suggested time limits per section and ask that you do not spend more than six hours in total.

Background

At Better Collective (BC), one of the ways we generate revenue is by referring our customers to sportsbook operators. This marketing model is the main focus of our Affiliate Analytics department. To accomplish this, we display ads on our various owned and operated websites (such as Action Network and VegasInsider) that direct users to sportsbooks. These operators pay us when BC referred customers create accounts and deposit money.
For this assessment, you will be working with a dataset from one of our partner sportsbooks. This is real data that represents exactly the type of data you will be working with. However, it is sampled for this exercise. The data shows betting activity over time for customers Better Collective referred to the sportsbook. The data is aggregated by player and month, which means there should be one row of data for every month a player is active (ie deposited money or placed a bet.)

For this assessment, you will be working with a dataset from one of our partner sportsbooks. This is real data that represents exactly the type of data you will be working with. However, it is **sampled** for this exercise. The data shows betting activity over time for customers Better Collective referred to the sportsbook. The data is aggregated by player and month, which means there should be one row of data for every month a player is active (ie deposited money or placed a bet.)

The data has the following fields:

- activity_month: month of activity
- account_id: anonymous player id
- brand_id: state
- reg_date: date the player signed up / registered
- ftd_date: date the player first deposited money (FTD = first time deposit)
- qp_date: date the player first placed a cash bet (QP = qualified player)

● ben_login_id: grouping that tells us which of our brands the player came from
● tracker_id: parameter in the ad URL that tells us exactly where on our site the player originated
● player_reg_product: product the player first played
● total_deposit: how much money the player deposited
● total_handle: how much money the player wagered on sports bets and casino games
● total_ngr: how much revenue did the sportsbook make on this player activity

The field **total_ngr** (net gaming revenue) represents how much money the sportsbook makes off the customer and is our best representation of a player's value. Ultimately, the goal of our department is to determine CLVs for our players. We do not need a full CLV model for this test. Instead, we will focus on the first part of understanding CLV, churn.

Hint: You will need to create a new calculated field called months_active to help you with your churn model. This should be the difference (in months) between activity_month and ftd_date. For example, when the month of the activity_month is equal to the month of the ftd_date then months_active = 0.

Challenge Sections

Section 1: Data Exploration (~1 hour)

1. Exploratory Data Analysis:
    - ○ Provide the code you use to explore the data including any notes on statistics or trends that you find interesting
    - ○ Are there any obvious issues with the data? What challenges do you foresee based on this initial exploration?

Section 2: Predictive Modeling (2 hours)

Build Two Churn Models

1. Implement a baseline model that predicts churn for months 0 to 60. You can choose any algorithm you find appropriate
    - ○ Provide the code used to train and evaluate the model as well as brief notes on why you chose the model you did
    - ○ The goal is to see how players churn out over time
2. Build a model that classifies or predicts the likelihood of whether or not a customer (or segment of customers) will churn.
    - ○ Provide the code used to train and evaluate the model as well as brief notes on why you chose the model you did
    - ○ The goal is to predict early on the likelihood a player will churn out quickly and therefore have a low CLV

Notes:
● Include all the work you did to prepare the data for modeling, including handling missing values (if necessary), feature engineering, splitting the data

into training and test sets, etc.

- Include all assumptions you have made for your models

Hint: For your second model, you will need to make assumptions on how you are defining churn. For example, a player is not necessarily fully churned / low value just because they miss one month of activity. Be clear about the thresholds you are using For your definition.

Section 3: Production (~3 hour)

1. Model Evaluation:
    - How would you evaluate the effectiveness of this model over time as new data continues to come in?
    - What alternatives would you consider if you have more data and time?
2. Production & Scale:
    - How would you approach the fact that this model needs to be repeated approximately 2000 times for different partners and geographies combinations?
    - Where would you save the outputs of the models? In which format? Providing that is going to be a batch run
    - Once you have your training code, prepare an additional .py that uses principles of OOP to deliver the model as an artifact

Submission

Please submit the following:

1. A git repository with an adequate structure as if you were delivering it to an MLOps team (you don't actually need to create YAMLs or additional files, but show how you would structure the project in github)

(Note - your answers do not need to be formal sentences or paragraphs. Short thoughts and bullets are great.)

We look forward to reviewing your submission. Good luck!