**MACHINE LEARNING DESIGN INTERVIEW**

**Machine Learning Primer**

**I.Feature Selection and Feature Engineering**

**1. Giới thiệu (Introduction)**

● **Mục tiêu của Feature Selection và Feature Engineering:** Nêu rõ mục tiêu của việc lựa chọn đặc trưng và kỹ thuật đặc trưng trong việc xây dựng mô hình học máy hiệu quả.

**● Tầm quan trọng:** Giải thích tại sao việc lựa chọn đặc trưng và kỹ thuật đặc trưng lại quan trọng đối với hiệu suất của mô hình.

**2. Xử lý dữ liệu phân loại (Handling Categorical Data)**

**● Mã hóa One-Hot (One-Hot Encoding):**

○ **Định nghĩa:** Giải thích kỹ thuật mã hóa one-hot và cách thức hoạt động.

○ **Ưu điểm:** Nêu rõ ưu điểm của phương pháp này, chẳng hạn như dễ thực hiện và hiệu quả với các đặc trưng phân loại có số lượng giá trị vừa phải.

**○ Nhược điểm:**

**■ Mô hình cây:** Giải thích tại sao mô hình dựa trên cây (decision tree, random forest, boosted tree) không hoạt động tốt với mã hóa one-hot, đặc biệt khi cây có nhiều cấp.

**■ Tính toán và bộ nhớ:** Mô tả vấn đề về tính toán và bộ nhớ khi sử dụng mã hóa one-hot với các đặc trưng có số lượng giá trị lớn.

**○ Lựa chọn tốt nhất (Best Practices):** Đề xuất cách xử lý khi số lượng giá trị phân loại lớn, chẳng hạn như nhóm các giá trị ít xuất hiện vào một nhóm "Khác". Đảm bảo pipeline xử lý được dữ liệu chưa gặp trong tập kiểm tra.

**○ Ví dụ:** Cung cấp ví dụ về cách thực hiện mã hóa one-hot trong Python, sử dụng pandas.get\_dummies hoặc sklearn OneHotEncoder. Lưu ý sự khác biệt giữa hai thư viện và cách xử lý dữ liệu mới trong tập kiểm tra.

**○ Ứng dụng thực tế:** Nêu ví dụ về việc sử dụng mã hóa one-hot trong các công ty công nghệ, chẳng hạn như Uber.5

**● Mã hóa trung bình (Mean Encoding):**

○ **Định nghĩa:** Giải thích kỹ thuật mã hóa trung bình và cách thức hoạt động.

**○ Ưu điểm:**

■ **Giảm chiều:** Mô tả cách mã hóa trung bình giúp giảm số lượng đặc trưng so với mã hóa one-hot.

■ **Hiệu suất:** Nêu rõ những trường hợp mã hóa trung bình có thể hoạt động tốt hơn mã hóa one-hot.

**○ Nhược điểm:**

■ **Rò rỉ nhãn (Label Leakage):** Giải thích vấn đề rò rỉ nhãn và cách xử lý bằng cách sử dụng dữ liệu riêng biệt để tính toán mã hóa trung bình.

■ **Tăng cường tính mạnh mẽ:** Mô tả cách áp dụng làm mịn cộng hoặc phương pháp kiểm tra chéo để tăng cường tính mạnh mẽ cho mã hóa trung bình.

○ **Ví dụ:** Cung cấp ví dụ cụ thể về cách thực hiện mã hóa trung bình, chẳng hạn như sử dụng tập dữ liệu thu nhập người trưởng thành.

**● Băm đặc trưng (Feature Hashing):**

○ **Định nghĩa:** Giải thích kỹ thuật băm đặc trưng và cách thức hoạt động.

**○ Ưu điểm:**

■ **Giảm chiều:** Nêu rõ cách băm đặc trưng giúp giảm số lượng đặc trưng, đặc biệt là với các đặc trưng có số lượng giá trị rất lớn.

■ **Tiết kiệm bộ nhớ:** Mô tả lợi ích về bộ nhớ của băm đặc trưng.

**○ Nhược điểm:**

■ **Va chạm (Collision):** Giải thích vấn đề va chạm trong băm đặc trưng và ảnh hưởng của nó đến hiệu suất mô hình.

■ **Khó giải thích:** Nêu rõ nhược điểm về khả năng giải thích của băm đặc trưng.

○ **Lựa chọn hàm băm:** Liệt kê các hàm băm thường được sử dụng, chẳng hạn như MurmurHash3, Jenkins, CityHash và MD5.

○ **Ví dụ:** Cung cấp ví dụ cụ thể về cách thực hiện băm đặc trưng, chẳng hạn như chuyển đổi văn bản "the quick brown fox" thành vectơ đặc trưng.

○ **Ứng dụng thực tế:** Nêu ví dụ về việc sử dụng băm đặc trưng trong các công ty công nghệ, chẳng hạn như Booking, Facebook, Yahoo, Yandex, Avazu và Criteo.

**● Đặc trưng chéo (Cross Feature):**

**○ Định nghĩa:** Giải thích khái niệm đặc trưng chéo và cách tạo ra đặc trưng chéo từ hai hoặc nhiều đặc trưng phân loại.

**○ Ưu điểm:**

■ **Nắm bắt tương tác:** Mô tả cách đặc trưng chéo giúp nắm bắt tương tác giữa các đặc trưng khác nhau.

**■ Cải thiện hiệu suất:** Nêu rõ những trường hợp đặc trưng chéo có thể cải thiện hiệu suất của mô hình.

**○ Nhược điểm:**

■ **Tăng số lượng đặc trưng:** Giải thích vấn đề về việc tăng số lượng đặc trưng khi sử dụng đặc trưng chéo.

○ **Kết hợp với băm đặc trưng:** Mô tả cách kết hợp đặc trưng chéo với băm đặc trưng để giảm số lượng đặc trưng.

○ **Ví dụ:** Cung cấp ví dụ cụ thể về cách sử dụng đặc trưng chéo, chẳng hạn như trong bài toán dự đoán nhu cầu của Uber dựa trên dữ liệu vị trí đón khách.

○ **Ứng dụng thực tế:** Nêu ví dụ về việc sử dụng đặc trưng chéo trong các hệ thống đề xuất.

**3. Nhúng thực thể (Entity Embedding):**

**● Định nghĩa**: Giải thích khái niệm nhúng thực thể và cách biểu diễn các đối tượng phân loại dưới dạng vectơ số thực.

**● Ưu điểm:**

**○ Giữ ý nghĩa ngữ nghĩa:** Mô tả cách nhúng thực thể giúp giữ ý nghĩa ngữ nghĩa của các đặc trưng phân loại.

**○ Hiệu suất:** Nêu rõ những trường hợp nhúng thực thể có thể hoạt động tốt hơn các kỹ thuật mã hóa khác.

**● Cách huấn luyện nhúng:**

**○ Nhúng được đào tạo trước (Pre-trained Embedding):**

■ **Định nghĩa:** Giải thích kỹ thuật nhúng được đào tạo trước và cách thức hoạt động.

■ **Ví dụ:** Cung cấp ví dụ về nhúng từ (word embedding) sử dụng Word2Vec.

■ **Ứng dụng thực tế:** Nêu ví dụ về việc sử dụng nhúng được đào tạo trước trong các công ty công nghệ, chẳng hạn như Instagram, Pinterest.

**○ Đồng huấn luyện (Co-trained Embedding):**

**■ Định nghĩa:** Giải thích kỹ thuật đồng huấn luyện và cách thức hoạt động.

**■ Ví dụ:**

**●** Mô tả cách YouTube huấn luyện nhúng trong thành phần truy xuất bằng cách sử dụng kiến trúc hai tháp.

● Mô tả cách LinkedIn huấn luyện nhúng bằng kiến trúc kim tự tháp ngược.

● Mô tả cách Pinterest huấn luyện nhúng trực quan cho tìm kiếm hình ảnh.

**■ Ứng dụng thực tế:** Nêu ví dụ về việc sử dụng đồng huấn luyện trong các công ty công nghệ, chẳng hạn như Twitter, YouTube, DoorDash.

**● Đánh giá chất lượng nhúng:**

**○ Thực hiện nhiệm vụ:** Đánh giá chất lượng nhúng dựa trên hiệu suất của mô hình trong các nhiệm vụ cụ thể.

**○ Trực quan hóa:** Mô tả cách sử dụng các kỹ thuật trực quan hóa như t-SNE hoặc UMAP để đánh giá chất lượng nhúng.

**● Kích thước nhúng:**

**○ Quy tắc ngón tay cái:** Giới thiệu quy tắc ngón tay cái để lựa chọn kích thước nhúng dựa trên số lượng giá trị phân loại.

**○ Điều chỉnh siêu tham số:** Mô tả cách điều chỉnh kích thước nhúng như một siêu tham số để tối ưu hóa hiệu suất mô hình.

**4. Xử lý dữ liệu số (Handling Numerical Data)**

**● Chuẩn hóa (Normalization):**

**○ Định nghĩa:** Giải thích kỹ thuật chuẩn hóa và mục đích của nó.

**○ Công thức:** Cung cấp công thức chuẩn hóa dữ liệu số về phạm vi [-1, 1](hoặc1).

**○ Vấn đề ngoại lai:** Nêu rõ vấn đề ngoại lai có thể ảnh hưởng đến kết quả chuẩn hóa.

**○ Giải pháp cắt (Clipping):** Đề xuất cách sử dụng kỹ thuật cắt để xử lý ngoại lai trong quá trình chuẩn hóa.

**● Chuẩn hóa (Standardization):**

**○ Định nghĩa:** Giải thích kỹ thuật chuẩn hóa và mục đích của nó.

**○ Công thức:** Cung cấp công thức chuẩn hóa dữ liệu số về phân phối chuẩn.

**○ Phân phối luật lũy thừa:** Mô tả cách xử lý dữ liệu số có phân phối luật lũy thừa bằng cách sử dụng biến đổi logarit**.**

**5. Tóm tắt và kết luận (Summary and Conclusion):**

**● Tóm tắt các kỹ thuật:** Tóm tắt ngắn gọn các kỹ thuật lựa chọn đặc trưng và kỹ thuật đặc trưng đã được thảo luận.

**● Lựa chọn kỹ thuật phù hợp:** Cung cấp hướng dẫn về cách lựa chọn kỹ thuật phù hợp dựa trên loại dữ liệu và mục tiêu của bài toán.

**● Tầm quan trọng:** Nhấn mạnh lại tầm quan trọng của việc lựa chọn đặc trưng và kỹ thuật đặc trưng trong việc xây dựng mô hình học máy hiệu quả.

**II.Training Pipeline**

**1. Giới thiệu (Introduction):**

● **Mục tiêu của Training Pipeline:** Nêu rõ mục tiêu của việc xây dựng một training pipeline hiệu quả trong việc huấn luyện mô hình học máy với dữ liệu lớn.

● **Tầm quan trọng của Training Pipeline:** Giải thích tại sao việc thiết kế và tối ưu hóa training pipeline lại quan trọng đối với hiệu suất và khả năng mở rộng của mô hình.

● **Các thành phần chính:** Liệt kê các thành phần chính của một training pipeline, bao gồm:

○ **Lưu trữ dữ liệu:** Nêu các định dạng dữ liệu phù hợp cho việc lưu trữ dữ liệu lớn, chẳng hạn như Parquet, Avro, ORC và tfrecord. Giải thích ưu điểm của các định dạng này trong việc xử lý dữ liệu lớn cho các bài toán học máy và phân tích.

○ **Phân vùng dữ liệu:** Mô tả kỹ thuật phân vùng dữ liệu và lợi ích của nó trong việc xử lý dữ liệu lớn. Ví dụ: phân vùng dữ liệu theo năm và tháng.

○ **Xử lý dữ liệu mất cân bằng:** Giải thích vấn đề mất cân bằng lớp (class imbalance) và ảnh hưởng của nó đến hiệu suất mô hình. Đề xuất các giải pháp như:

■ **Lấy mẫu quá mức (Oversampling):** Mô tả kỹ thuật lấy mẫu quá mức cho lớp thiểu số.

■ **Lấy mẫu thiếu mức (Undersampling):** Mô tả kỹ thuật lấy mẫu thiếu mức cho lớp đa số.

■ **Gán trọng số (Weighting):** Giải thích kỹ thuật gán trọng số cho các mẫu dữ liệu để điều chỉnh sự mất cân bằng lớp.

○ **Tạo dữ liệu:** Thảo luận về các chiến lược tạo dữ liệu cho các bài toán học máy, đặc biệt là trong các trường hợp thiếu dữ liệu nhãn. Cung cấp ví dụ về cách LinkedIn và Facebook tạo dữ liệu cho các hệ thống đề xuất.

○ **Chia dữ liệu huấn luyện/kiểm tra:** Nhấn mạnh tầm quan trọng của việc chia dữ liệu huấn luyện/kiểm tra phù hợp với bài toán cụ thể. Ví dụ: trong các bài toán dự báo, cần phải chia dữ liệu theo thứ tự thời gian.

○ **Huấn luyện lại mô hình:** Giải thích nhu cầu huấn luyện lại mô hình trong môi trường sản xuất và các kỹ thuật huấn luyện lại khác nhau, bao gồm:

■ **Huấn luyện lại từ đầu:** Mô tả kỹ thuật huấn luyện lại mô hình từ đầu với dữ liệu mới.

■ **Huấn luyện lại từng phần:** Mô tả kỹ thuật huấn luyện lại một phần mô hình với dữ liệu mới.

■ **Huấn luyện lại khởi động nóng (Warm-start Retraining):** Mô tả kỹ thuật huấn luyện lại khởi động nóng, trong đó mô hình được khởi tạo bằng trọng số từ mô hình đã huấn luyện trước đó.

**2. Xử lý mất cân bằng lớp (Handling Imbalance Class Distribution):**

● **Vấn đề:** Giải thích chi tiết vấn đề mất cân bằng lớp trong các bài toán như phát hiện gian lận, dự đoán nhấp chuột hoặc phát hiện thư rác. Ví dụ: trong dự đoán nhấp chuột quảng cáo, tỷ lệ chuyển đổi thường rất thấp (khoảng 0,2%).

● **Ảnh hưởng:** Mô tả ảnh hưởng của mất cân bằng lớp đến hiệu suất mô hình, chẳng hạn như mô hình tập trung quá nhiều vào việc học các mẫu âm tính.

● **Giải pháp:** Cung cấp ví dụ và giải thích chi tiết về các giải pháp đã đề cập ở phần giới thiệu:

○ **Lấy mẫu quá mức:** Ví dụ: sao chép các mẫu dương tính nhiều lần.

○ **Lấy mẫu thiếu mức:** Ví dụ: loại bỏ ngẫu nhiên các mẫu âm tính.

○ **Gán trọng số:** Ví dụ: tăng trọng số cho các mẫu dương tính trong hàm mất mát. Cung cấp công thức tính toán trọng số cho các mẫu dương tính dựa trên tỷ lệ mất cân bằng.

**3. Chiến lược tạo dữ liệu (Data Generation Strategy):**

● **Vấn đề:** Giải thích thách thức trong việc thu thập dữ liệu nhãn, đặc biệt là trong các bài toán học có giám sát.

● **Giải pháp:** Cung cấp các ví dụ cụ thể về cách các công ty công nghệ như LinkedIn và Facebook tạo dữ liệu cho các hệ thống đề xuất:

○ **LinkedIn Feed Ranking:** Sử dụng thứ tự thời gian để tạo dữ liệu nhãn ban đầu.3

○ **Facebook Place Recommendation:** Sử dụng dữ liệu lượt thích của người dùng làm nhãn dương tính và lấy mẫu các địa điểm khác làm nhãn âm tính.

● **Ví dụ chi tiết:**

○ **LinkedIn Course Recommendation:** Mô tả chi tiết cách LinkedIn tạo dữ liệu cho hệ thống đề xuất khóa học, bao gồm:

■ **Vấn đề khởi động nguội (Cold Start Problem):** Giải thích thách thức khi thiếu dữ liệu tương tác người dùng ban đầu.

■ **Sử dụng khảo sát người dùng:** Mô tả cách sử dụng khảo sát người dùng trong quá trình onboarding để thu thập dữ liệu nhãn ban đầu.

■ **Sử dụng kỹ năng làm thước đo mức độ liên quan:** Giải thích ý tưởng sử dụng kỹ năng để đo lường mức độ liên quan giữa người học và khóa học.

■ **Mô hình dựa trên kỹ năng (Skill-Based Model):** Mô tả mô hình dựa trên kỹ năng, bao gồm hai thành phần chính:

● **Khóa học đến kỹ năng (Course to Skill):**

○ **Gắn thẻ thủ công (Manual Tagging):** Sử dụng phân loại học để gắn thẻ khóa học với kỹ năng.

○ **Tận dụng tính năng gắn thẻ kỹ năng của LinkedIn (Leverage LinkedIn Skill Taggers):** Sử dụng tính năng gắn thẻ kỹ năng tự động của LinkedIn để trích xuất kỹ năng từ dữ liệu khóa học.

○ **Mô hình học có giám sát:** Huấn luyện mô hình phân loại để dự đoán mức độ liên quan giữa khóa học và kỹ năng.

○ **Học bán giám sát (Semi-Supervised Learning):** Sử dụng kỹ thuật học bán giám sát để cải thiện hiệu suất mô hình.

● **Thành viên đến kỹ năng (Member to Skill):**

○ **Sử dụng thông tin hồ sơ:** Trích xuất kỹ năng từ hồ sơ người dung.

○ **Sử dụng chức danh và ngành nghề:** Sử dụng thông tin chức danh và ngành nghề để suy luận kỹ năng của người dùng.

○ **Kết hợp trọng số:** Kết hợp thông tin từ hồ sơ và chức danh/ngành nghề để tạo ánh xạ kỹ năng chính xác hơn.

**4. Chia dữ liệu huấn luyện/kiểm tra (How to Split Train/Test Data):**

● **Tầm quan trọng:** Nhấn mạnh tầm quan trọng của việc chia dữ liệu huấn luyện/kiểm tra phù hợp trong môi trường sản xuất.

● **Bài toán dự báo:** Giải thích tại sao cần phải chia dữ liệu theo thứ tự thời gian trong các bài toán dự báo. Ví dụ: không nên sử dụng dữ liệu trong tương lai để dự đoán dữ liệu trong quá khứ.

● **Cửa sổ mở rộng (Expanding Window):** Mô tả kỹ thuật cửa sổ mở rộng để chia dữ liệu huấn luyện/kiểm tra trong các bài toán dự báo.

**5. Yêu cầu huấn luyện lại (Retraining Requirements):**

● **Vấn đề:** Giải thích tại sao cần phải huấn luyện lại mô hình trong môi trường sản xuất, đặc biệt là khi phân phối dữ liệu thay đổi theo thời gian.

● **Tần suất huấn luyện lại:** Thảo luận về tần suất huấn luyện lại mô hình và các yếu tố cần xem xét, chẳng hạn như tốc độ thay đổi dữ liệu và chi phí tính toán.

● **Cân bằng giữa độ phức tạp của mô hình và thời gian huấn luyện:** Nhấn mạnh tầm quan trọng của việc cân bằng giữa độ phức tạp của mô hình và thời gian huấn luyện trong quá trình thiết kế training pipeline.

● **Các cấp độ huấn luyện lại:** Giới thiệu bốn cấp độ huấn luyện lại mô hình và giải thích sự khác biệt giữa chúng:

○ **Huấn luyện lại ngoại tuyến (Offline Retraining):** Huấn luyện lại mô hình theo lịch trình cố định.

○ **Huấn luyện lại trực tuyến (Online Retraining):** Huấn luyện lại mô hình liên tục dựa trên dữ liệu mới nhất.

○ **Huấn luyện lại theo lịch trình (Scheduled Retraining):** Huấn luyện lại mô hình theo lịch trình linh hoạt dựa trên các yếu tố như hiệu suất mô hình và thay đổi dữ liệu.7

○ **Huấn luyện lại khởi động nóng:** Huấn luyện lại một phần mô hình với dữ liệu mới, sử dụng trọng số từ mô hình đã huấn luyện trước đó.

**Ngoài ra, bạn có thể bổ sung thêm các nội dung sau:**

● **Xử lý dữ liệu bị thiếu:** Thảo luận về các kỹ thuật xử lý dữ liệu bị thiếu, chẳng hạn như thay thế bằng giá trị trung bình hoặc sử dụng mô hình để dự đoán giá trị bị thiếu.

● **Kiểm tra chéo:** Giải thích kỹ thuật kiểm tra chéo và lợi ích của nó trong việc đánh giá hiệu suất mô hình.

● **Tối ưu hóa siêu tham số:** Thảo luận về các kỹ thuật tối ưu hóa siêu tham số, chẳng hạn như tìm kiếm ngẫu nhiên hoặc tìm kiếm theo lưới.

● **Công cụ và framework:** Giới thiệu các công cụ và framework phổ biến để xây dựng training pipeline, chẳng hạn như TensorFlow, PyTorch, Apache Spark và Scikit-learn.

**III.Loss Function and Metrics Evaluation**

**1. Giới thiệu (Introduction):**

● **Tầm quan trọng của Hàm Mất mát và Đánh giá Metrics:** Bắt đầu bằng việc giải thích lý do tại sao việc lựa chọn hàm mất mát và xác định metrics phù hợp là rất quan trọng trong việc phát triển sản phẩm/dịch vụ học máy.

● **Phân loại bài toán:** Chia phần này thành hai phần chính: Hồi quy (Regression) và Phân loại (Classification).

**2. Hàm Mất mát Hồi quy (Regression Loss):**

● **Tổng quan:** Giới thiệu ngắn gọn về hàm mất mát hồi quy và mục đích của chúng trong việc huấn luyện mô hình.

● **Các hàm mất mát phổ biến:**

○ **Sai số bình phương trung bình (Mean Squared Error - MSE):**

■ **Công thức:** Cung cấp công thức tính MSE.

■ **Ưu điểm:** Đơn giản, dễ hiểu, dễ tính toán đạo hàm.

■ **Nhược điểm:** Nhạy cảm với các giá trị ngoại lai.

○ **Sai số tuyệt đối trung bình (Mean Absolute Error - MAE):**

■ **Công thức:** Cung cấp công thức tính MAE.

■ **Ưu điểm:** Ít nhạy cảm với các giá trị ngoại lai hơn MSE.

■ **Nhược điểm:** Đạo hàm không liên tục tại 0.

○ **Hàm mất mát Huber (Huber Loss):**

■ **Ý tưởng:** Giải thích ý tưởng đằng sau hàm mất mát Huber, kết hợp ưu điểm của MSE và MAE.

■ **Công thức:** Cung cấp công thức tính hàm mất mát Huber.

■ **Siêu tham số delta:** Giải thích vai trò của siêu tham số delta và cách điều chỉnh nó.

○ **Hàm mất mát phân vị (Quantile Loss):**

■ **Ứng dụng:** Giải thích khi nào nên sử dụng hàm mất mát phân vị, chẳng hạn như khi đánh giá thấp hoặc đánh giá cao có mức độ quan trọng khác nhau.

■ **Công thức:** Cung cấp công thức tính hàm mất mát phân vị.

■ **Tham số lambda:** Giải thích vai trò của tham số lambda và cách điều chỉnh nó.

● **Ví dụ minh họa:**

○ Cung cấp các ví dụ minh họa về cách tính MSE và MAE cho một tập dữ liệu cụ thể.

○ So sánh MSE và MAE trong trường hợp có giá trị ngoại lai.

● **Lựa chọn hàm mất mát:**

○ Nhấn mạnh rằng việc lựa chọn hàm mất mát phụ thuộc vào bài toán cụ thể và mục tiêu của mô hình.

○ Cung cấp các ví dụ về việc lựa chọn hàm mất mát trong các ứng dụng thực tế:

■ Uber sử dụng hàm mất mát pseudo-Huber và log-cosh để xấp xỉ hàm mất mát Huber và MAE trong huấn luyện XGBoost phân tán.

■ Doordash sử dụng MSE cho mô hình ETA (Estimated Time Arrival) sau đó chuyển sang hàm mất mát phân vị và MSE bất đối xứng tùy chỉnh.

**3. Hàm Mất mát Phân loại (Classification Loss):**

● **Tổng quan:** Giới thiệu ngắn gọn về hàm mất mát phân loại và mục đích của chúng trong việc huấn luyện mô hình.

● **Các hàm mất mát phổ biến:**

○ **Entropi chéo (Cross Entropy):**

■ **Công thức:** Cung cấp công thức tính entropi chéo.

■ **Ưu điểm:** Phổ biến, hiệu quả trong các bài toán phân loại nhị phân và đa lớp.

○ **Entropi chéo chuẩn hóa (Normalized Cross Entropy - NCE):**

■ **Vấn đề:** Giải thích vấn đề nhạy cảm với tỷ lệ chuyển đổi nền (background CTR) của entropi chéo trong dự đoán nhấp chuột quảng cáo.

■ **Giải pháp:** Giới thiệu NCE như một giải pháp cho vấn đề này.

■ **Công thức:** Cung cấp công thức tính NCE.

■ **Ưu điểm:** Ít nhạy cảm với tỷ lệ chuyển đổi nền, có thang đo trực quan.

■ **Ví dụ minh họa:** Cung cấp ví dụ về việc so sánh hai mô hình dự đoán nhấp chuột quảng cáo với entropi chéo và NCE.

○ **Hàm mất mát Focal (Focal Loss):**

■ **Ý tưởng:** Giải thích ý tưởng đằng sau hàm mất mát Focal, tập trung vào các mẫu khó phân loại.

■ **Công thức:** Cung cấp công thức tính hàm mất mát Focal.

■ **Tham số gamma:** Giải thích vai trò của tham số gamma và cách điều chỉnh nó.

■ **Ứng dụng:** Cung cấp ví dụ về việc sử dụng hàm mất mát Focal trong phát hiện đối tượng.

○ **Hàm mất mát Hinge (Hinge Loss):**

■ **Ý tưởng:** Giải thích ý tưởng đằng sau hàm mất mát Hinge, phạt các mẫu được phân loại sai và các mẫu được phân loại đúng nhưng nằm trong khoảng cách nhất định từ ranh giới quyết định.

■ **Công thức:** Cung cấp công thức tính hàm mất mát Hinge.

■ **Ví dụ minh họa:** Cung cấp ví dụ về cách tính hàm mất mát Hinge cho một tập dữ liệu cụ thể.

■ **Ứng dụng:** Cung cấp ví dụ về việc sử dụng hàm mất mát Hinge trong các ứng dụng thực tế:

● Airbnb sử dụng hàm mất mát Hinge trong thư viện học máy Aerosolve.

● Uber sử dụng hàm mất mát Hinge với GraphSAGE để đề xuất món ăn và nhà hàng cho Uber Eats.

**4. Đánh giá mô hình (Model Evaluation):**

● **Phân loại:** Chia phần này thành hai phần nhỏ: Metrics Offline và Metrics Online.

● **Metrics Offline:**

○ **Tổng quan:** Giới thiệu ngắn gọn về metrics offline và mục đích của chúng trong việc đánh giá mô hình.

○ **Các metrics phổ biến:**

■ **Log loss:**

● **Ưu điểm:** Phổ biến, dễ tính toán.

● **Nhược điểm:** Khó giải thích trực quan.

■ **MAE:**

● **Ưu điểm:** Dễ hiểu, ít nhạy cảm với các giá trị ngoại lai.

● **Nhược điểm:** Đạo hàm không liên tục tại 0.

■ **R2 (hệ số xác định):**

● **Ưu điểm:** Đo lường mức độ phù hợp của mô hình với dữ liệu.

● **Nhược điểm:** Không cho biết liệu mô hình có bị overfitting hay không.

○ **Diện tích dưới đường cong (Area Under the Curve - AUC):**

■ **Ý tưởng:** Giải thích ý tưởng đằng sau AUC và cách tính toán nó.

■ **Ưu điểm:** Đo lường hiệu suất tổng thể của mô hình phân loại nhị phân.

■ **Nhược điểm:** Không cho biết chi tiết về hiệu suất của mô hình ở các ngưỡng khác nhau.

○ **Precision và Recall:**

■ **Định nghĩa:** Giải thích rõ ràng precision và recall trong bài toán tìm kiếm thông tin.

■ **Precision tại k (P@K):** Giải thích P@K và cách tính toán nó.

■ **Độ chính xác trung bình (Average Precision - AP):** Giải thích AP và cách tính toán nó.

■ **Độ chính xác trung bình tại n (AP@n):** Giải thích AP@n và cách tính toán nó.

○ **Xếp hạng trung bình nghịch đảo (Mean Reciprocal Rank - MRR):**

■ **Ý tưởng:** Giải thích MRR và cách tính toán nó.

■ **Ưu điểm:** Dễ hiểu, dễ tính toán.

■ **Nhược điểm:** Chỉ xem xét kết quả liên quan đầu tiên.

○ **Lợi ích giảm giá tích lũy chuẩn hóa (Normalized Discounted Cumulative Gain - NDCG):**

■ **Ý tưởng:** Giải thích ý tưởng đằng sau NDCG và cách tính toán nó.

■ **Ưu điểm:** Xem xét thứ tự của các kết quả liên quan.

■ **Nhược điểm:** Phức tạp hơn MRR.

● **Metrics Online:**

○ **Tổng quan:** Giới thiệu ngắn gọn về metrics online và mục đích của chúng trong việc đánh giá mô hình.

○ **Các metrics phổ biến:**

■ **Tăng trưởng doanh thu (Revenue Lift):** Đo lường sự thay đổi phần trăm của doanh thu.

■ **Tỷ lệ nhấp (Click-Through Rate - CTR):** Đo lường tỷ lệ người dùng nhấp vào liên kết hoặc quảng cáo.

■ **Tỷ lệ chuyển đổi (Conversion Rate):** Đo lường tỷ lệ người dùng thực hiện hành động mong muốn, chẳng hạn như mua hàng hoặc đăng ký.

○ **Kiểm tra A/B:** Giải thích tầm quan trọng của kiểm tra A/B trong việc đánh giá hiệu suất mô hình trong môi trường sản xuất.

○ **Giám sát:** Nhấn mạnh tầm quan trọng của việc giám sát hiệu suất mô hình và phát hiện các vấn đề như overfitting hoặc data drift.

**5. Các vấn đề đặc biệt (Special Considerations):**

● **Độ lệch vị trí (Position Bias):**

○ **Vấn đề:** Giải thích vấn đề độ lệch vị trí trong đề xuất quảng cáo.

○ **Ảnh hưởng:** Mô tả ảnh hưởng của độ lệch vị trí đến việc huấn luyện mô hình.

○ **Giải pháp:**

■ **Sử dụng vị trí làm đặc trưng:** Giải thích cách sử dụng vị trí làm đặc trưng trong huấn luyện mô hình và loại bỏ độ lệch vị trí trong quá trình phục vụ.

■ **Điểm số xu hướng nghịch đảo (Inverse Propensity Score - IPS):** Giới thiệu IPS như một phương pháp khác để loại bỏ độ lệch vị trí.

● **Hiệu chỉnh (Calibration):**

○ **Định nghĩa:** Giải thích hiệu chỉnh là gì và tầm quan trọng của nó trong đánh giá mô hình.

○ **Công thức:** Cung cấp công thức tính hiệu chỉnh.

○ **Khi nào hiệu chỉnh quan trọng:** Giải thích khi nào hiệu chỉnh là quan trọng, chẳng hạn như trong các hệ thống quan trọng đòi hỏi dự đoán chính xác về giá trị thực tế.

○ **Ví dụ minh họa:** Cung cấp ví dụ về việc so sánh hai mô hình phân loại nhị phân với và không có hiệu chỉnh.

○ **Biểu đồ hiệu chỉnh (Calibration Plot):** Giải thích biểu đồ hiệu chỉnh và cách sử dụng nó để đánh giá hiệu chỉnh của mô hình.

○ **Phương pháp hiệu chỉnh:** Giới thiệu các phương pháp hiệu chỉnh phổ biến, chẳng hạn như phương pháp isotonic và phương pháp Platt.

● **Bài toán không tĩnh (Nonstationary Problem):**

○ **Vấn đề:** Giải thích vấn đề dữ liệu thay đổi theo thời gian trong môi trường trực tuyến.

○ **Giải pháp:** Giới thiệu Hồi quy Logistic Bayes (Bayesian Logistic Regression) như một phương pháp xử lý vấn đề này.

● **Khám phá so với Khai thác (Exploration vs. Exploitation):**

○ **Vấn đề:** Giải thích vấn đề cân bằng giữa khám phá và khai thác trong đề xuất quảng cáo.

○ **Giải pháp:** Giới thiệu Lấy mẫu Thompson (Thompson Sampling) như một kỹ thuật cân bằng giữa khám phá và khai thác.

**6. Bài tập ôn tập (Interview Exercises):**

● Cung cấp một số bài tập ôn tập để giúp người đọc củng cố kiến thức về hàm mất mát và đánh giá metrics.

● Ví dụ:

○ Thiết kế hệ thống học máy để dự đoán số người sẽ tham dự một sự kiện trên Facebook.

○ Thiết kế mô hình học máy để phát hiện xem hệ thống phát hiện đối tượng người có thực sự phát hiện con người ngoài đời hay con người trên TV/poster.

**IV. Common Sampling Techniques**

**Lấy mẫu Ngẫu nhiên (Random Sampling)**

Đây là kỹ thuật đơn giản nhất, trong đó chúng ta chọn ngẫu nhiên các mẫu từ tập dữ liệu lớn1. Kỹ thuật này hiệu quả trong nhiều ứng dụng Machine Learning, ví dụ như chọn video chưa xem từ kho video để làm mẫu âm cho mô hình nhúng người dùng.

**Lấy mẫu Từ chối (Rejection Sampling)**

Kỹ thuật này được sử dụng khi cần tạo mẫu ngẫu nhiên từ một phân phối phức tạp. Ví dụ, để tạo hàm rand10() (tạo số nguyên ngẫu nhiên từ 1 đến 10) từ hàm rand7() (tạo số nguyên ngẫu nhiên từ 1 đến 7), ta có thể sử dụng kỹ thuật lấy mẫu từ chối.

Các yếu tố cần thiết cho kỹ thuật này bao gồm:

● Phân phối đề xuất (proposal density) q(z): dễ lấy mẫu hơn phân phối gốc p(z).

● Phân phối gốc p(z): khó lấy mẫu nhưng có thể ước lượng theo hằng số tỷ lệ.

● Hằng số k: sao cho kq(z) >= p(z) với mọi giá trị z.

Kỹ thuật này được sử dụng trong thực tế, ví dụ như Uber sử dụng nó trong mô hình tạo văn bản của họ3.

**Lấy mẫu Có trọng số (Weight Sampling)**

Khi mỗi mục trong tập dữ liệu có trọng số riêng, xác suất chọn mỗi mục phụ thuộc vào trọng số đó. Bài toán này được gọi là lấy mẫu ngẫu nhiên có trọng số. Bạn có thể tham khảo mã Python minh họa cho kỹ thuật này và tìm hiểu thêm về lấy mẫu có trọng số trên luồng dữ liệu.

**Lấy mẫu Quan trọng (Importance Sampling)**

Kỹ thuật này được sử dụng để ước lượng giá trị kỳ vọng của một hàm h(x) khi việc lấy mẫu trực tiếp từ phân phối gốc p(x) là khó khăn hoặc không hiệu quả. Thay vào đó, ta lấy mẫu từ phân phối thay thế q(x) và điều chỉnh kết quả bằng tỷ lệ p(x)/q(x).

**Lấy mẫu Phân tầng (Stratified Sampling)**

Kỹ thuật này giúp bảo toàn phân phối dữ liệu gốc bằng cách chia tập dữ liệu thành các phân tầng và lấy mẫu từ mỗi phân tầng.

**Lấy mẫu Hồ chứa (Reservoir Sampling)**

Kỹ thuật này được sử dụng để chọn k mẫu từ tập dữ liệu n mẫu, trong đó n thường rất lớn hoặc chưa biết trước.

**V. Common Deep Learning Model Architecture**

**Kiến trúc Wide and Deep**

Kiến trúc này kết hợp hai mô hình: Wide (linear model) và Deep (deep neural network), nhằm tận dụng ưu điểm của cả hai.

● Mô hình Wide: Học các mối tương quan tuyến tính giữa các đặc trưng, giúp ghi nhớ (memorization) các mẫu xuất hiện thường xuyên trong dữ liệu lịch sử.

● Mô hình Deep: Học các biểu diễn ẩn phức tạp, giúp khái quát hóa (generalization) và khám phá các kết hợp đặc trưng mới.

Lợi ích:

● Kết hợp khả năng ghi nhớ và khái quát hóa, cải thiện hiệu suất mô hình.

● Được Google sử dụng thành công trong đề xuất ứng dụng trên Google Play Store.

**Kiến trúc Two-Tower**

Kiến trúc này sử dụng hai mạng nơ-ron riêng biệt để học biểu diễn cho truy vấn (query) và mục (item), thường được dùng trong hệ thống truy xuất thông tin.

● Tháp trái: Mã hóa thông tin người dùng và ngữ cảnh (context).

● Tháp phải: Mã hóa thông tin mục.

Điểm số tương đồng giữa truy vấn và mục được tính bằng tích vô hướng (dot product) của hai biểu diễn.

Lợi ích:

● Khả năng mở rộng cho tập dữ liệu lớn với hàng triệu mục.

● Giảm thiểu vấn đề "đuôi dài" (long-tail problem), khi dữ liệu huấn luyện thưa thớt cho hầu hết các mục.

Ví dụ:

● YouTube sử dụng kiến trúc này trong hệ thống đề xuất video "Xem tiếp theo" (Watch Next).

● Facebook sử dụng kiến trúc tương tự trong mô hình đề xuất DLRM mã nguồn mở.

**Kiến trúc Deep Cross Network (DCN)**

Kiến trúc này được thiết kế để học các đặc trưng chéo (cross features) một cách hiệu quả, đặc biệt hữu ích cho các ứng dụng web với nhiều đặc trưng thưa thớt.

● Mạng Cross: Áp dụng phép nhân chéo (feature crossing) ở mỗi tầng, bậc đa thức tăng dần theo độ sâu của tầng.

● Mạng Deep: Mạng nơ-ron truyền thống (MLP).

Lợi ích:

● Tự động học các đặc trưng chéo, cải thiện hiệu suất so với kiến trúc DL truyền thống.

● Được Google sử dụng trong các ứng dụng thực tế, mang lại cải thiện đáng kể về AUC.

**Kiến trúc Multitask Learning**

Kiến trúc này huấn luyện một mô hình cho nhiều nhiệm vụ (tasks) đồng thời, chia sẻ các tầng dưới (bottom layers) để tận dụng biểu diễn đã học.

Lợi ích:

● Tận dụng kiến thức từ các nhiệm vụ khác, cải thiện hiệu suất cho từng nhiệm vụ.

● Huấn luyện một mô hình duy nhất cho nhiều nhiệm vụ, giảm thiểu chi phí và phức tạp.

Ví dụ:

● YouTube sử dụng kiến trúc này trong hệ thống đề xuất video "Xem tiếp theo", dự đoán đồng thời xác suất xem, thích, và bình luận cho video.

● Instagram kết hợp xác suất thích và bình luận để xếp hạng nội dung.

Lưu ý:

● Việc lựa chọn kiến trúc phù hợp phụ thuộc vào bài toán cụ thể và đặc điểm của dữ liệu.

● Các kiến trúc này có thể được kết hợp và tùy chỉnh để đáp ứng nhu cầu của từng ứng dụng.

**VI.** **A/B Testing Fundamental**

A/B testing là một kỹ thuật thử nghiệm so sánh để đánh giá hiệu suất của hai phiên bản (A và B) của một tính năng hoặc sản phẩm. Phiên bản A là phiên bản hiện tại (được gọi là phiên bản kiểm soát), trong khi phiên bản B là phiên bản mới với những thay đổi hoặc cải tiến.

Trong Machine Learning, A/B testing thường được sử dụng để:

● Đánh giá hiệu suất của các mô hình hoặc thuật toán Machine Learning mới.

● So sánh hiệu quả của các kỹ thuật xử lý dữ liệu, kỹ thuật tạo đặc trưng, hoặc các tham số mô hình khác nhau.

● Đo lường tác động của các thay đổi đối với trải nghiệm người dùng và các chỉ số kinh doanh quan trọng.

Các bước thực hiện A/B Testing:

1. Xác định mục tiêu: Rõ ràng về mục tiêu của thử nghiệm, ví dụ như tăng tỷ lệ nhấp (CTR), thời gian xem video, hoặc tỷ lệ chuyển đổi.

2. Chọn chỉ số đánh giá: Xác định các chỉ số phù hợp để đo lường hiệu suất, ví dụ như CTR, thời gian xem, tỷ lệ chuyển đổi, hoặc doanh thu.

3. Tạo hai phiên bản (A/B): Phiên bản A là phiên bản kiểm soát, phiên bản B là phiên bản mới.

4. Phân chia ngẫu nhiên người dùng: Chia người dùng thành hai nhóm ngẫu nhiên, một nhóm tiếp xúc với phiên bản A và nhóm còn lại tiếp xúc với phiên bản B.

5. Thu thập dữ liệu: Theo dõi và ghi lại dữ liệu về hành vi người dùng trong mỗi nhóm.

6. Phân tích kết quả: So sánh hiệu suất của hai phiên bản dựa trên các chỉ số đã chọn.

7. Rút ra kết luận: Xác định phiên bản nào hiệu quả hơn và đưa ra quyết định triển khai.

Lưu ý quan trọng khi thực hiện A/B testing:

● Đảm bảo tính ngẫu nhiên: Việc phân chia người dùng vào các nhóm A/B phải ngẫu nhiên để giảm thiểu sai lệch.

● Kiểm soát các yếu tố nhiễu: Các yếu tố bên ngoài có thể ảnh hưởng đến kết quả thử nghiệm cần được kiểm soát hoặc loại bỏ.

● Thời gian thử nghiệm: Thời gian thử nghiệm đủ dài để thu thập đủ dữ liệu và đảm bảo kết quả đáng tin cậy.

● Ý nghĩa thống kê: Sử dụng các phương pháp thống kê để đánh giá ý nghĩa của sự khác biệt về hiệu suất giữa hai phiên bản.

Ví dụ A/B testing trong các nguồn:

● Nguồn mô tả cách SageMaker và LinkedIn thực hiện A/B testing để đánh giá hiệu quả của các mô hình mới trong môi trường sản xuất.

● Nguồn minh họa cách LinkedIn sử dụng kỹ thuật giảm giá hiển thị (impression discount) trong tính năng "Những người bạn có thể biết" (PYMK) và sử dụng A/B testing để tối ưu hóa tỷ lệ chuyển đổi.

Kết luận:

A/B testing là một công cụ mạnh mẽ để đánh giá hiệu quả của các thay đổi và cải tiến trong Machine Learning. Ứng dụng A/B testing giúp ra quyết định dựa trên dữ liệu và tối ưu hóa hiệu suất của các mô hình và sản phẩm.

Thông tin bổ sung:

● Các nguồn cung cấp thông tin chi tiết về các khía cạnh khác nhau của A/B testing, bao gồm thiết kế thử nghiệm, phân tích dữ liệu, và rút ra kết luận.

● Các nguồn cũng đề cập đến các kỹ thuật nâng cao trong A/B testing, như phân bổ ngân sách (budget-splitting) và kiểm soát các yếu tố nhiễu.

**VII.** **Common Deployment Patterns**

**Phân vùng Dữ liệu (Data Partitioning):**

● Dữ liệu thường được phân vùng theo thời gian (ví dụ: năm, tháng) để tối ưu hóa hiệu suất truy vấn và huấn luyện song song/phân tán.

● Các định dạng dữ liệu cột (columnar) như Parquet, Avro, ORC giúp truy vấn nhanh hơn và tiết kiệm chi phí so với CSV.

**Xử lý Dữ liệu Mất cân bằng (Imbalance Class Distribution):**

● Các kỹ thuật như **upsampling**, **downsampling**, và **gán trọng số** được sử dụng để xử lý dữ liệu mất cân bằng, phổ biến trong các bài toán như phát hiện gian lận, dự đoán nhấp chuột.

● **Downsampling** giúp giảm kích thước dữ liệu huấn luyện và thời gian huấn luyện.

● **Gán trọng số** cho các mẫu dương giúp mô hình tập trung hơn vào việc học các mẫu hiếm.

**Chiến lược Tạo Dữ liệu (Data Generation Strategy):**

● **Thu thập dữ liệu ngầm (implicit labels):** Sử dụng các tín hiệu như lượt xem, lượt nhấp để tạo nhãn cho mô hình giám sát.

● **Giải quyết bài toán khởi động nguội (cold start):** Sử dụng các kỹ thuật như gán nhãn thủ công, khai thác dữ liệu hiện có, hoặc xây dựng mô hình giám sát để tạo nhãn cho dữ liệu mới.

**Chia Dữ liệu Huấn luyện/Kiểm tra (Split Train/Test Data):**

● **Tôn trọng thứ tự thời gian:** Trong các bài toán dự báo hoặc phụ thuộc thời gian, cần chia dữ liệu huấn luyện/kiểm tra theo thứ tự thời gian để tránh rò rỉ thông tin từ tương lai.

● **Kỹ thuật cửa sổ mở rộng (expanding window):** Sử dụng để đánh giá hiệu suất mô hình trên dữ liệu mới theo thời gian.

**Yêu cầu Huấn luyện lại (Retraining Requirements):**

● **Phân phối dữ liệu thay đổi:** Phân phối dữ liệu trong thực tế thường thay đổi theo thời gian, do đó mô hình cần được huấn luyện lại thường xuyên để duy trì hiệu suất.

● **Các mức độ huấn luyện lại:**

○ **Mức 0:** Huấn luyện một lần và không huấn luyện lại.

○ **Mức 1:** Huấn luyện lại định kỳ trên toàn bộ dữ liệu.

○ **Mức 2:** Huấn luyện lại từng phần của mô hình trên dữ liệu trực tuyến.

○ **Mức 3:** Huấn luyện lại các thành phần được cá nhân hóa trên dữ liệu cụ thể cho mỗi khóa.

**Kiến trúc Mô hình Deep Learning Phổ biến (Common Deep Learning Model Architectures):**

● **Wide and Deep:** Kết hợp mô hình tuyến tính (wide) và mạng nơ-ron sâu (deep) để tận dụng cả khả năng ghi nhớ và khái quát hóa.

● **Two-Tower:** Sử dụng hai mạng nơ-ron riêng biệt để học biểu diễn cho truy vấn (query) và mục (item), thường được dùng trong hệ thống truy xuất thông tin.

● **Deep Cross Network (DCN):** Tự động học các đặc trưng chéo (cross features) một cách hiệu quả, đặc biệt hữu ích cho các ứng dụng web với nhiều đặc trưng thưa thớt.

**Phục vụ Mô hình (Model Serving):**

● **Phục vụ ngoại tuyến (offline serving):** Tính toán trước các kết quả hoặc biểu diễn (ví dụ: nhúng) và lưu trữ chúng để truy xuất nhanh chóng trong quá trình suy luận.

● **Phục vụ trực tuyến (online serving):** Triển khai mô hình trên máy chủ để xử lý yêu cầu theo thời gian thực.

● **Tìm kiếm láng giềng gần đúng (Approximate Nearest Neighbor Search):** Sử dụng các kỹ thuật như **FAISS** để tìm kiếm các mục tương tự một cách hiệu quả trong không gian nhúng.

**Triển khai Ví dụ (Deployment Example):**

● Nguồn cung cấp một ví dụ về triển khai hệ thống xếp hạng quảng cáo (ad ranking system), bao gồm các thành phần như cơ sở dữ liệu, kho lưu trữ mô hình, dịch vụ truy xuất, dịch vụ xếp hạng, và bộ nhớ đệm đặc trưng.

● **Tách biệt dịch vụ truy xuất và xếp hạng:** Giúp tối ưu hóa hiệu suất và khả năng mở rộng của hệ thống.

● **Cập nhật đặc trưng theo thời gian thực:** Sử dụng các dịch vụ như Amazon DynamoDB để lưu trữ và cập nhật các đặc trưng thay đổi thường xuyên.

**Xử lý Sai lệch Huấn luyện-Phục vụ (Training-Serving Skew):**

● **Vấn đề:** Sự khác biệt về cách xử lý đặc trưng trong quá trình huấn luyện và phục vụ có thể dẫn đến sai lệch và giảm hiệu suất mô hình.

● **Giải pháp:** Sử dụng cùng một quy trình xử lý đặc trưng cho cả huấn luyện và phục vụ hoặc sử dụng ghi nhật ký đặc trưng (feature logging) để giám sát và phát hiện sai lệch.

**Tổng quan :** Phần này đã trình bày các mô hình triển khai phổ biến trong Machine Learning, bao gồm các kỹ thuật xử lý dữ liệu, kiến trúc mô hình, phương pháp phục vụ, và xử lý sai lệch. Việc lựa chọn mô hình triển khai phù hợp phụ thuộc vào yêu cầu cụ thể của từng ứng dụng, bao gồm hiệu suất, khả năng mở rộng, và độ phức tạp.

**VIII. Spotify: one simple mistake took four months to detect**

Phần này phân tích cách Spotify xây dựng "Mô hình Podcast" - dự đoán podcast mà người nghe có khả năng nghe trong mục "Chương trình bạn có thể thích". Bài học rút ra từ trường hợp này là **sai lệch huấn luyện-phục vụ**, một vấn đề phổ biến trong triển khai Machine Learning, đặc biệt khi các nhóm khác nhau chịu trách nhiệm cho huấn luyện và phục vụ mô hình.

**Mô tả vấn đề:**

● **Quy trình huấn luyện:** Đọc dữ liệu thô, thực hiện chuyển đổi đặc trưng, huấn luyện mô hình và lưu mô hình vào Kho lưu trữ mô hình.

● **Phục vụ trực tuyến:** Dịch vụ đề xuất (chạy trên cơ sở hạ tầng khác), chuyển đổi đặc trưng và đưa ra dự đoán.

● **Vấn đề phát sinh:** Chuyển đổi đặc trưng trong Phục vụ trực tuyến được thực hiện khác với quy trình huấn luyện

**Hậu quả:**

● Dữ liệu đầu vào khác nhau trong quá trình huấn luyện và phục vụ, dẫn đến vấn đề **sai lệch huấn luyện-phục vụ**.

● Mặc dù lỗi triển khai chỉ là một vài dòng mã, nhưng nó ảnh hưởng nghiêm trọng đến đề xuất của mô hình.

● Vấn đề này đã xảy ra trong bốn tháng trước khi được phát hiện.

**Giải pháp của Spotify:**

Spotify đã chọn giải pháp **ghi nhật ký đặc trưng** để giải quyết vấn đề này.

● **Triển khai chuyển đổi đặc trưng bằng Java.**

● **Ghi nhật ký các đặc trưng đã được chuyển đổi ở giai đoạn phục vụ.** Điều này rất quan trọng vì dịch vụ thượng nguồn được sử dụng cho chuyển đổi đặc trưng thuộc sở hữu của các nhóm khác nhau. Việc ghi nhật ký đặc trưng ở dịch vụ thượng nguồn sẽ gây khó khăn cho nhóm Machine Learning trong việc sở hữu và thực hiện thay đổi.3

● **Nhật ký đặc trưng sau đó được sử dụng để huấn luyện.**

● **Sử dụng Tensorflow Data Validation (TFDV) để so sánh sơ đồ dữ liệu huấn luyện và phục vụ, cũng như phân phối đặc trưng một cách thường xuyên.**

● **Thiết lập cảnh báo để phát hiện trôi dạt đặc trưng (sử dụng chỉ số khoảng cách Chebyshev).**

**Bài học kinh nghiệm:**

● Vấn đề sai lệch huấn luyện-phục vụ có thể rất khó phát hiện và có thể ảnh hưởng nghiêm trọng đến hiệu suất mô hình.

● Ghi nhật ký đặc trưng và giám sát phân phối đặc trưng là những kỹ thuật quan trọng để phát hiện và ngăn chặn sai lệch huấn luyện-phục vụ.

● Các nhóm Machine Learning cần có quyền kiểm soát quy trình xử lý đặc trưng cho cả huấn luyện và phục vụ.

**IX. Chapter Exercises**

**Quiz 1: Quiz on Cross Entropy : C**

**Quiz 2:** Log loss là một tên gọi khác của Cross Entropy. Bộ phân loại nào có giá trị Cross Entropy thấp hơn thì có hiệu suất tốt hơn.

Từ câu hỏi 1, ta đã tính được Cross Entropy của Classifier 1 là 0.6073375041358446.

Áp dụng cách tính tương tự, ta có thể tính được Cross Entropy của Classifier 2 là 0.6931471805599453.

**Kết luận:** Classifier 1 có categorical cross entropy loss tốt hơn.

**Đáp án:** (B)

**Quiz 3:** Để tính độ chính xác, ta cần biết số lượng dự đoán đúng trên tổng số dự đoán. Tuy nhiên, đề bài không cung cấp thông tin về ngưỡng phân loại để xác định dự đoán đúng hay sai.

Ví dụ: nếu ngưỡng là 0.6, một mẫu có pApple = 0.7 sẽ được dự đoán là lớp "Apple". Ngược lại, nếu ngưỡng là 0.8, mẫu này sẽ được dự đoán là lớp khác.

**Kết luận:** Không đủ thông tin để xác định bộ phân loại nào có độ chính xác cao hơn.

**Đáp án:** (C)

**Quiz 4 :** Hàm mất mát trong học sâu cần phải khả vi để có thể sử dụng thuật toán lan truyền ngược để cập nhật trọng số của mô hình.

Độ chính xác, được tính bằng số lượng dự đoán đúng trên tổng số dự đoán, là một hàm bước, không khả vi.

**Kết luận:** Độ chính xác không khả vi nên không thể được sử dụng làm hàm mất mát trong học sâu.

**Đáp án:** (B)