Target-free 会話型推薦における対話と推薦結果の特徴に基づく失敗予測

原田　悠真†　酒井　哲也†

† 早稲田大学基幹理工学研究科 情報理工・情報通信専攻　〒 169-8555 東京都新宿区大久保 3-4-1
E-mail: †yuuma6280@akane.waseda.jp, †tetsuyasakai@acm.org

あらまし　会話型推薦システムにおいて, 推薦の失敗を事前に検知することはユーザ満足度の維持に不可欠である. 既存研究は, ユーザがあらかじめ特定の正解アイテムを念頭に置く Target-biased なデータセットに依存しており, 実世界でユーザが対話を通じて好みを探索するシナリオを十分に反映していない. また, 予測モデルがユーザの発話特徴または推薦結果のいずれか一方のみに依存し, 両者の相互作用を捉えきれていない. 本研究では, Target-free な会話データを用い, ユーザの発話特徴とシステムの推薦結果特徴を統合的に利用する失敗予測手法を提案し, その有効性を検証した. 実験の結果, 提案手法は既存のベースライン手法と比較して AUC-ROC で最大 22.5%の改善を達成し, ユーザの期待とシステムの応答のミスマッチを検出することが, 失敗予測において重要な役割を果たすことが示された.

キーワード　会話型推薦，失敗予測，推薦システム

1 導　　入

近年, 会話型推薦システム (Conversational Recommender Systems; CRS) は, ユーザとの自然言語対話を通じて好みを理解し, 適切なアイテムを推薦する技術として注目を集めている. 従来の推薦システムと比較して, CRS はユーザの曖昧な要求や潜在的な嗜好を対話的に明確化できる点で優れており, 様々なドメインでの応用が期待されている.

しかし, CRS において全ての推薦が成功するわけではない. システムがユーザの意図を誤解したり, 不適切なアイテムを推薦したりすることで, ユーザ満足度の低下やシステムからの離脱につながる可能性がある. こうした推薦の失敗を事前に検知し適切な対応策をとることは, 実用的な CRS の構築において重要である. 推薦が失敗すると予測される場合, システムは追加の質問を行う, 推薦を一時保留する, あるいは説明を強化するといった介入が可能となり, ユーザ体験の改善につながる.

推薦の失敗予測に関する既存研究は一定の成果を上げているものの, いくつかの重要な課題が残されている. 第一に, 既存研究の多くは Target-biased なデータセット, すなわちユーザがあらかじめ特定の正解アイテムを念頭に置いて対話を行うという仮定に基づいている. 実世界では, ユーザは明確な目標を持たずに対話を開始し, システムとのやり取りを通じて徐々に好みを形成・探索することが多い. このような Target-free なシナリオにおける失敗予測は, 既存手法では十分に対応できていない. 第二に, 既存の予測モデルはユーザの発話特徴またはシステムの推薦結果特徴のいずれか一方に依存する傾向がある. Vlachou [6] は各会話ターンにおける推薦結果の一貫性に着目した失敗予測手法を提案した. また, Cai ら [1] はユーザの発話からユーザ意図 (user intent) を予測するとともに, 談話構造, 感情, 文脈といった多様な特徴カテゴリを定義し, ユーザ満足度を予測する手法を提案した. これらの研究では発話の特徴と推薦結果の特徴のいずれかに注目しており, 両者の相互作用を包括的に捉えきれていない. 推薦の成否は, ユーザの期待と実際に提示されたアイテムとの適合性によって決まるため, 両方の情報を統合的に活用することが不可欠である.

本研究では, これらの課題に対処するため, Kim ら [4] が提案した Pepper ユーザシミュレータを用いて生成された Target-free な会話データを用いた推薦失敗予測手法を提案する. 本研究は以下の Research Question に答えることを目的とする.

RQ: Target-free な会話型推薦において, 対話内容と推薦結果の関係性が失敗予測に寄与するか

この主要な問いに答えるため, ユーザの発話特徴とシステムの推薦結果特徴の相互作用に着目し, 以下の 2 つの観点から分析を行う.

RQ1: 感情極性と推薦結果の変化度の関係は失敗予測に寄与するか

RQ2: ユーザ要求の具体性と推薦結果の多様性の関係は失敗予測に寄与するか

これらの Research Question に基づき, ユーザの発話から抽出される特徴とシステムが推薦するアイテムの特徴を統合的に利用するモデルを構築し, 推薦が失敗する可能性を事前に予測する. 実験の結果, 提案手法は既存のベースライン手法と比較して AUC-ROC で最大 22.5%の改善を達成し, ユーザの期待とシステムの応答のミスマッチを捉えることが失敗予測に寄与することが示された.

2 関 連 研 究

2.1 会話型推薦における失敗予測

会話型推薦システムにおける失敗予測は, ユーザ体験の向上とシステムの効率的な運用において重要な研究課題である. 既存研究では, 主に推薦結果の特徴やユーザの発話特徴に基づく予測手法が提案されている. Vlachou [6] は, 会話型画像推薦タスクにおける失敗予測の問題を, 情報検索における Query Performance Prediction (QPP) の枠組みから着想を得て定式

化した. 彼らは, システムがターゲットアイテムを見つけられないシステム失敗と, ターゲットアイテムがカタログに存在しないカタログ失敗という 2 つの推薦シナリオを区別し, 複数ターンにわたる検索アイテムの埋め込み表現に含まれる意味情報を用いて会話の失敗を検知する予測器を提案した. 彼らの手法は, Autoencoder ベースの予測器が訓練ターンの上位検索アイテムの圧縮表現を学習し, 分類ラベルを用いて評価ターンを予測するものである. しかし, この研究は主に推薦結果の一貫性に焦点を当てており, ユーザの発話内容との相互作用は十分に考慮されていない.

一方, Cai ら [1] は, ユーザの発話背後にある意図とシステムの推薦に対する満足度を予測することがマルチターン対話ベースの会話型推薦システムの開発において重要であると指摘した. 彼らは, コンテンツ, 談話構造, 感情, 文脈といった多様なカテゴリの特徴を定義し, 様々な機械学習手法を比較することで, ユーザの意図と満足度を予測する手法を提案した. 実験の結果, ユーザ意図とシステムアクションの両方を含む対話行動特徴を活用することで, ユーザ満足度予測において良好な結果が得られることを示した. しかし, この研究はユーザの発話特徴に主眼を置いており, 推薦されるアイテムの特徴との関係性については限定的である.

これらの既存研究は, 推薦結果の特徴またはユーザの発話特徴のいずれか一方に依存する傾向があり, 両者の相互作用を包括的に捉える手法は確立されていない. また, これらの研究の多くは Target-biased なデータセット, すなわちユーザが事前に特定のターゲットアイテムを念頭に置いているという仮定に基づいており, 実世界のユーザが対話を通じて動的に好みを形成するシナリオに対応した失敗予測手法の開発が求められている.

2.2 会話型推薦システムのデータセット

会話型推薦システムの研究において, 適切なデータセットの構築は基盤となる重要な課題である. 既存のデータセットは, その収集方法や対話の性質によって大きく異なる特徴を持つ.

Li ら [5] が提案した ReDial (Recommendation Dialogues) は, 会話型推薦研究において広く利用されている代表的なデータセットである. ReDial は映画推薦を対象としており, クラウドソーシングによって収集された人間同士の自然な対話を含む. このデータセットでは, 推薦者 (Recommender) と被推薦者 (Seeker) の 2 つの役割を持つ参加者が, 実際の映画について対話を行う. ReDial の特徴は, 対話が自然言語で記録されているだけでなく, 各発話に言及された映画のメタデータ (タイトル, ジャンル, 評価など) が紐付けられている点にある. しかし, 人同士の会話データのため, 会話型推薦のタスクには不十分である.

Wu ら [7] が提案した Fashion IQ は, ファッションドメインにおける画像検索と自然言語対話を融合させたデータセットで, 複雑な視覚的特徴を持つアイテムを対象とするため, 会話型推薦の評価に広く用いられている. Fashion IQ の特徴は, 基準となる画像（Candidate）に対し, ユーザが「より袖が長いもの」「色が濃いもの」といった自然言語による修正指示（Relative Caption）を与えることで, ターゲットとなる画像を絞り込んでいく点にある. このデータセットは Target-biased な性質を持ち, ユーザが明確な目標アイテムを念頭に置いて対話を行うシナリオを想定している.

この Target-bias の問題および, 会話型推薦に適した会話データを作成するため, Kim ら [4] は Pepper ユーザシミュレータを提案した. Pepper は, Target-free な会話データを生成できる点で既存手法と異なる. このシミュレータは, ユーザが特定の目標アイテムを持たない状態から対話を開始し, 推薦システムとの相互作用を通じて好みを動的に形成するプロセスをモデル化している. Pepper は大規模言語モデル (Large Language Model; LLM) を基盤とし, ユーザの潜在的な嗜好プロファイルと対話履歴に基づいて, 各ターンでの応答を生成する. 生成される対話には, ユーザの満足度や受容性といったラベルが付与されており, 推薦の成否を分析するための情報が含まれている.

Pepper によって生成されたデータセットは, 従来の Target-biased なデータセットでは捉えられなかった探索的な対話シナリオにおける推薦の失敗パターンを分析することを可能にする. ユーザが明確な目標を持たない状況では, 推薦の成否を判断する基準も動的に変化し, システムの対応もより柔軟性が求められる. このような Target-free なシナリオにおける失敗予測は, 実用的な会話型推薦システムの構築において重要な研究課題である. 本研究では, この Pepper シミュレータによって生成されたデータを用いることで, より現実的な対話環境における推薦失敗予測手法の提案を目指す.

3 方 法 論

本節では Target-free な会話型推薦において, ユーザの発話特徴とシステムの推薦結果特徴の相互作用が失敗予測に寄与するという仮説のもと, ユーザの意図と推薦結果のミスマッチに関連する 2 つの新しい特徴量を説明する. 各 Research Question に対応する特徴量の設計と抽出方法について述べる.

3.1 RQ1: 感情極性と推薦結果の変化度に基づくミスマッチ特徴量

推薦の失敗は, ユーザの期待と実際に提示された推薦結果とのズレから生じることが多い. ユーザが肯定的な感情を示しているにもかかわらず, システムが推薦内容を大きく変更する場合, ユーザの好みを正しく理解できていない可能性がある. 逆に, ユーザが否定的な感情を示しているにもかかわらず, システムが推薦内容をほとんど変更しない場合も, ユーザのフィードバックを適切に反映できていないことを意味する.

このような感情とシステム応答の不整合を捉えるため, 以下のようなミスマッチ特徴量を提案する.

$$M_1 = -(s_{u_t} \cdot (2 \cdot \mathrm{sim}(r_t, r_{t-1}) - 1)) \qquad (1)$$

ここで, s_{u_t} はターン t におけるユーザ発話の感情極性スコア, $\mathrm{sim}(r_t, r_{t-1})$ はターン t の推薦結果 r_t と前ターン $t-1$ の推薦結果 r_{t-1} の内容類似度を表す. 以下, 各要素の定義について説明する.

3.1.1 感情極性スコア

ユーザ発話の感情極性 $s_{u_t} \in [-1, 1]$ は, 推薦に対するユーザの満足度や受容性を反映する重要な指標である. $s_{u_t} = 1$ は完全に肯定的, $s_{u_t} = -1$ は完全に否定的, $s_{u_t} = 0$ は中立を表す.

感情極性の抽出には, 感情分析モデルである DistilBERT[1] を使用する. このモデルは各発話に対して POSITIVE または NEGATIVE のラベルと信頼度スコア $p \in [0, 1]$ を出力する. これを $[-1, 1]$ の範囲に変換するため, 以下の式を用いる:

$$s_{u_t} = \begin{cases} 2p - 1 & \text{(POSITIVE)} \\ 1 - 2p & \text{(NEGATIVE)} \end{cases} \tag{2}$$

例えば, POSITIVE ラベルで信頼度 0.9 の場合, $s_{u_t} = 2 \times 0.9 - 1 = 0.8$ となり, 強い肯定を示す. 一方, NEGATIVE ラベルで信頼度 0.9 の場合, $s_{u_t} = 1 - 2 \times 0.9 = -0.8$ となり, 強い否定を示す.

3.1.2 推薦内容類似度

推薦内容類似度 $\mathrm{sim}(r_t, r_{t-1}) \in [0, 1]$ は, 連続する 2 つのターン間でシステムの推薦がどの程度変化したかを測る指標である. 値が 1 に近いほど推薦内容の変化が小さく, 0 に近いほど大きく変化したことを示す.

単純な映画タイトルの一致度ではなく, 映画の内容的類似性を考慮するため, 本研究では 5 つの属性 (ジャンル, あらすじ, 監督, 脚本家, 出演者) を統合した類似度を計算する.

3.1.3 ミスマッチスコアの解釈

式 (1) で定義されたミスマッチ特徴量 M_1 は, 感情極性と推薦内容の変化度の関係性を定量化する. この式の設計意図を以下で説明する.

まず, $(2 \cdot \mathrm{sim}(r_t, r_{t-1}) - 1) \in [-1, 1]$ は, 推薦内容類似度を中心化した値である. $\mathrm{sim}(r_t, r_{t-1}) = 1$(推薦内容が不変) のとき $+1$, $\mathrm{sim}(r_t, r_{t-1}) = 0$(完全に異なる推薦) のとき -1, $\mathrm{sim}(r_t, r_{t-1}) = 0.5$(中程度の変化) のとき 0 となる.

この中心化された類似度と感情極性の積に負号を付けることで, 以下のような解釈が可能となる:

適切な応答 (負のミスマッチ)

ユーザが肯定的 ($s_{u_t} > 0$) で推薦内容が維持される (sim ≈ 1) 場合, または, ユーザが否定的 ($s_{u_t} < 0$) で推薦内容が大きく変化する (sim ≈ 0) 場合, M_1 は負の値をとる. これはシステムがユーザのフィードバックを適切に反映していることを示す.

不適切な応答 (正のミスマッチ)

ユーザが肯定的 ($s_{u_t} > 0$) だが推薦内容が大きく変化する (sim ≈ 0) 場合, または, ユーザが否定的 ($s_{u_t} < 0$) だが推薦内容が維持される (sim ≈ 1) 場合, M_1 は正の値をとる. これはユーザの期待とシステムの応答に不整合があることを示し, 推薦失敗の可能性が高まると考えられる.

具体例として, ユーザが「好みじゃない」と否定的な発話 ($s_{u_t} = -0.8$) をした後に, システムが全く異なるジャンルの映画を推薦 (sim $= 0.1$) した場合を考える. このとき,

$$M_1 = -(-0.8 \cdot (2 \times 0.1 - 1)) = -(-0.8 \times (-0.8)) = -0.64$$

となり, 負のミスマッチ, すなわち適切な応答として評価される. 一方, 同じ否定的発話の後に類似した映画を推薦 (sim $= 0.9$) した場合,

$$M_1 = -(-0.8 \cdot (2 \times 0.9 - 1)) = -(-0.8 \times 0.8) = 0.64$$

となり, 正のミスマッチ, すなわち不適切な応答として評価される.

3.2 RQ2: ユーザ要求の具体性と推薦結果の多様性に基づく特徴量

Target-free な会話型推薦において, ユーザは明確な目標を持たずに対話を開始し, システムとのやり取りを通じて徐々に好みを探索・形成する. このプロセスにおいて, ユーザの要求がどの程度具体的になっているか, そしてシステムがどの程度推薦の多様性を変化させるかは, 推薦の成否に大きく影響する. ユーザの要求が具体的になっているにもかかわらず, システムが推薦をより多様化させる場合, ユーザは焦点を絞れず混乱する可能性がある. 逆に, ユーザの要求が漠然としたままであるにもかかわらず, システムが推薦の多様性を減少させる場合, 探索の機会が制限される.

このようなユーザ要求の具体性変化と推薦結果の多様性変化の不整合を捉えるため, 以下のような特徴量を提案する.

$$M_2 = \Delta\mathrm{spec}_t \cdot \Delta\mathrm{div}_t \tag{3}$$

ここで, $\Delta\mathrm{spec}_t$ はターン t におけるユーザ発話の具体性の変化量, $\Delta\mathrm{div}_t$ はターン t の推薦結果の多様性の変化量を表す. 以下, 各要素の定義について説明する.

3.2.1 ユーザ要求の具体性スコア

ユーザ発話の具体性 $\mathrm{spec}(u_t) \in [0, 1]$ は, ユーザがどの程度明確かつ詳細な要求を示しているかを表す指標である. $\mathrm{spec}(u_t) = 1$ は非常に具体的な要求, $\mathrm{spec}(u_t) = 0$ は非常に漠然とした要求を示す.

具体性を測るため, 本研究では以下の 3 つの指標を統合する:

a) 語彙の豊富さ

発話内のユニークな単語数を総単語数で割った値. 語彙が豊富であるほど, ユーザは詳細な情報を提供していると考えられる:

$$\mathrm{unique\_ratio} = \frac{|\text{unique words}|}{|\text{total words}|} \tag{4}$$

b) 固有名詞の数

発話に含まれる固有名詞の数. 映画タイトル, 俳優名, 監督名などの固有名詞が多いほど, 要求が具体的であると考えられる. 固有名詞の抽出には, BERT-Large[2] をベースとした CoNLL-2003 データセットで事前学習された固有表現認識 (Named Entity Recognition; NER) モデルを使用する. 検出された固有名詞の

1：https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english

2：https://huggingface.co/dbmdz/bert-large-cased-finetuned-conll03-english

数を最大 10 個として正規化する:

$$\text{named\_entity\_score} = \min\left(\frac{|\text{entities}|}{10}, 1.0\right) \tag{5}$$

c） 文の長さ

発話の単語数．長い発話ほど詳細な情報を含む傾向がある．最大 100 語として正規化する:

$$\text{length\_score} = \min\left(\frac{|\text{words}|}{100}, 1.0\right) \tag{6}$$

3.2.2 推薦結果の多様性スコア

推薦結果の多様性 $\text{div}(r_t) \in [0,1]$ は, ターン t で提示された映画群がどの程度多様であるかを表す指標である．$\text{div}(r_t) = 1$ は非常に多様な推薦, $\text{div}(r_t) = 0$ は非常に類似した推薦を示す．

多様性を測るため, 本研究では映画の 5 つの属性 (ジャンル, あらすじ, 監督, 脚本家, 出演者) について, 以下の指標を計算する:

a） ジャンルの多様性

推薦された映画群のジャンル分布のエントロピーを用いて多様性を測る．R_t をターン t で推薦された映画の集合, G をそれらの映画が持つ全ジャンルの集合とすると, 各ジャンル $g \in G$ の出現確率 $p(g)$ を計算し, エントロピー H を以下のように求める:

$$H = -\sum_{g \in G} p(g) \log_2 p(g) \tag{7}$$

エントロピーを正規化するため, 最大エントロピー $H_{\max} = \log_2 |G|$ で割る:

$$\text{genre\_diversity} = \frac{H}{H_{\max}} \tag{8}$$

エントロピーが高いほど, ジャンルが均等に分散しており, 多様性が高いことを示す．

b） 監督・脚本家・出演者の多様性

それぞれの属性について, ユニークな人物の数を総人物数で割った値を用いる．D, W, S をそれぞれ監督, 脚本家, 出演者の集合とすると:

$$\text{director\_diversity} = \frac{|\text{unique directors}|}{|D|} \tag{9}$$

$$\text{writer\_diversity} = \frac{|\text{unique writers}|}{|W|} \tag{10}$$

$$\text{star\_diversity} = \frac{|\text{unique stars}|}{|S|} \tag{11}$$

重複が少ないほど, 多様性が高いと評価される．

c） あらすじの多様性

推薦された映画群のあらすじ間のペアワイズ非類似度の平均を用いる．各映画 i のあらすじを TF-IDF ベクトル $\mathbf{v}_i$ で表現し, すべての映画ペア (i, j) について類似度を計算する:

$$\text{sim}_{ij} = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|} \tag{12}$$

平均非類似度を多様性として定義する:

$$\text{plot\_diversity} = 1 - \frac{1}{\binom{n}{2}} \sum_{i<j} \text{sim}_{ij} \tag{13}$$

ここで, n は推薦された映画の数, $\binom{n}{2} = \frac{n(n-1)}{2}$ は映画ペアの総数である．

3.2.3 具体性と多様性の変化量

対話の動的な変化を捉えるため, ターン間での具体性と多様性の変化量を計算する．

具体性の変化量 Δspec_t と多様性の変化量 Δdiv_t を以下のように定義する:

$$\Delta\text{spec}_t = \text{spec}(u_t) - \text{spec}(u_{t-1}) \tag{14}$$

$$\Delta\text{div}_t = \text{div}(r_t) - \text{div}(r_{t-1}) \tag{15}$$

$\Delta\text{spec}_t > 0$ は, ユーザの要求がより具体的になったことを示し, $\Delta\text{spec}_t < 0$ は, より漠然となったことを示す．同様に, $\Delta\text{div}_t > 0$ は, システムの推薦がより多様になったことを示し, $\Delta\text{div}_t < 0$ は, より絞り込まれたことを示す．

3.2.4 ミスマッチスコアの解釈

式 (3) で定義された特徴量 M_2 は, ユーザ要求の具体性変化と推薦結果の多様性変化の関係性を定量化する．この式の設計意図を以下で説明する．

両方の変化量の積により, 以下のような解釈が可能となる:

適切な応答 (負のミスマッチ)

一方の変化量が正で他方が負の場合, M_2 は負の値をとる．これは以下の 2 つのケースに対応する:

- ユーザの要求がより具体的になっている ($\Delta\text{spec}_t > 0$) のに, システムが推薦を絞り込む ($\Delta\text{div}_t < 0$) 場合．これは適切な応答である可能性が高い．
- ユーザの要求がより漠然となっている ($\Delta\text{spec}_t < 0$) のに, システムが推薦をより多様化させる ($\Delta\text{div}_t > 0$) 場合．これも, 探索を促す適切な応答である可能性が高い．

不適切な応答 (正のミスマッチ)

両方の変化量が正または両方が負の場合, M_2 は正の値をとる．これは以下の 2 つのケースに対応する:

- ユーザの要求がより具体的になり ($\Delta\text{spec}_t > 0$), システムも推薦をより多様化させる ($\Delta\text{div}_t > 0$) 場合．ユーザが具体化している際に多様性を増やすことは, 必ずしも適切ではない可能性がある．
- ユーザの要求がより漠然となり ($\Delta\text{spec}_t < 0$), システムも推薦を絞り込む ($\Delta\text{div}_t < 0$) 場合．これも, ユーザが探索段階に戻っている際に選択肢を狭めることは適切でない可能性がある．

具体例として, ユーザが前のターンで「何か面白い映画はない?」($\text{spec}(u_{t-1}) = 0.2$) と言った後, 現在のターンで「アクション映画で, トム・クルーズが出演している作品が見たい」($\text{spec}(u_t) = 0.8$) と具体的な発話をした場合を考える．この場合, $\Delta\text{spec}_t = 0.8 - 0.2 = 0.6$ となる．

もしこのとき, システムが前のターンでは様々なジャンルの映画を推薦 ($\text{div}(r_{t-1}) = 0.9$) していたが, 現在のターンで

はアクション映画に絞った推薦 ($\mathrm{div}(r_t) = 0.3$) を行った場合, $\Delta\mathrm{div}_t = 0.3 - 0.9 = -0.6$ となる. このとき,

$$M_2 = 0.6 \times (-0.6) = -0.36$$

となり, 負のミスマッチが検出される. これは, ユーザの具体化に合わせてシステムが適切に推薦を絞り込んでおり, 適切な応答を示している.

一方, 同じ状況でシステムが推薦をさらに多様化させた場合 ($\mathrm{div}(r_t) = 0.95$, $\Delta\mathrm{div}_t = 0.05$),

$$M_2 = 0.6 \times 0.05 = 0.03$$

となり, 正のミスマッチが検出される. これは, ユーザが具体的な要求をしているにもかかわらず, システムが多様性を増やしており, ユーザの意図に応えていない可能性を示唆する.

4 評価実験

本節では, 提案手法の有効性を検証するために実施した評価実験について述べる. まず, 実験に使用したデータセットの詳細と前処理について説明し, 次に実験設定と評価指標について述べる. 最後に, 実験結果とその分析を示す.

4.1 データセット

4.1.1 Pepper ユーザシミュレータによるデータ生成

本研究では, Kim ら [4] が提案した Pepper ユーザシミュレータを用いて会話型推薦システムのデータを生成した. Pepper は, Target-free な会話型推薦のシナリオをシミュレートできる点で, 既存のデータセットと一線を画す. 従来の Target-biased なデータセットとは異なり, ユーザは特定の目標映画を持たずに対話を開始し, システムとの相互作用を通じて好みを動的に形成するプロセスがモデル化されている.

データ生成には, 会話型推薦システムとユーザシミュレータの両方に LLM である GPT-4o-mini[3]を使用した. 会話型推薦システムは, ユーザの発話を理解し, 映画データベースから適切な推薦を行う役割を担う. ユーザシミュレータは, 与えられたペルソナと視聴履歴に基づいて, 自然言語でシステムと対話を行う. 両者が LLM ベースであることにより, 実際のユーザとシステムの対話に近い, 多様で自然な会話データの生成が可能となっている.

各会話データは以下の要素から構成される:

ペルソナ (persona): ユーザの映画嗜好を記述したプロファイル. 好きな要素 (depth, dark humor, mystery など) と嫌いな要素 (overly cheesy, lack of closure など) が自然言語で記述されている.

視聴済み映画 (seen movies): ユーザが過去に視聴した映画のリスト. これらはユーザの嗜好を反映しており, 推薦の際に参考にされる.

目標映画 (target movies): システムが推薦すべき理想的な映画のリスト. ただし, Target-free な設定では, ユーザはこれらの映画を明示的に知らない状態で対話を開始する.

対話ターン: 各ターンは, ユーザの発話 (Seeker), 推薦者の応答 (Recommender), 推薦された映画リスト (top_k), およびマッチング情報 (match) を含む. 推薦者は各ターンで 4 個の映画を推薦する.

推薦の成否は, 以下の基準で判定される:

成功 (Success): target movies または seen movies に含まれる映画を少なくとも 1 つ推薦できた場合

失敗 (Failure): 対話が終了するまでに, 上記の映画を 1 つも推薦できなかった場合

4.1.2 データの前処理

生成された会話データに対して, 以下の前処理を施した:

a) 推薦成功の基準緩和

Target-free な設定において, 完全に一致する映画を推薦することは困難である. そこで, target movies および seen movies に含まれる映画と**類似した映画**を推薦した場合も成功とみなすように基準を緩和した.

両方の類似度が閾値 0.4 以上である場合, 2 つの映画を類似していると判定した. この処理により, 207 件の会話データが失敗から成功に再分類された.

b) 会話長の調整

予測タスクを適切に設定するため, 会話長に関して以下の調整を行った:

- **成功会話の切り詰め**: 推薦が成功した会話については, 成功直前のターンで会話を切り詰めた. これにより, モデルが「次のターンで失敗するかどうか」を予測するタスクとして定式化できる.
- **最小ターン数の設定**: 3 ターン未満の会話は, 特徴量の抽出が困難であるため除外した.
- **失敗会話のランダム切り詰め**: 失敗した会話については, 3 ターンから 8 ターンの間でランダムに切り詰めた. これにより, 成功会話と失敗会話の長さ分布を均等化し, モデルが会話長に過度に依存することを防ぐ.

4.1.3 データセットの統計

前処理後のデータセットは, 合計 2,417 件の会話データから構成される. 会話ターン数の平均は 4.95 である. 成功データと失敗データの分布は以下の通りである:

表 1 データセットの統計情報

項目	成功データ	失敗データ
ファイル数	394	2023
平均ターン数	4.70	5.00
中央値	5.0	5.0
最大ターン数	7	7
最小ターン数	3	3
標準偏差	1.35	1.42

成功データと失敗データのターン数分布を表 2 に示す. 成功データは比較的早期 (3〜5 ターン) に推薦が成功する傾向がある一方, 失敗データはより長いターン数にわたって分散して

3 : https://platform.openai.com/docs/models

いる.

表 2 ターン数分布

ターン数	成功データ		失敗データ	
	件数	割合	件数	割合
3 ターン	97	24.6%	413	20.4%
4 ターン	95	24.1%	391	19.3%
5 ターン	83	21.1%	408	20.2%
6 ターン	68	17.3%	399	19.7%
7 ターン	51	12.9%	412	20.4%

このデータセットは, 成功データと失敗データの比率が約 1:5 と不均衡である. この不均衡は, 実世界の会話型推薦システムにおける推薦の難しさを反映している.

4.2 実験設定

4.2.1 特徴量の抽出

各会話データから, 以下の 4 つの特徴量セットを抽出した:

Baseline 1 (感情極性ベース): 各ターン (ターン 1〜3) におけるユーザの発話をもとにした感情極性

Baseline 2 (推薦結果の一貫性ベース): 各ターン (ターン 1〜3) における推薦結果の一貫性スコア

Proposed RQ1: 各ターン (ターン 1〜3) における感情極性と推薦結果のミスマッチ

Proposed RQ2: 各ターン (ターン 1〜3) におけるユーザ発話の具体性と推薦結果の多様性のミスマッチ

4.2.2 モデルと学習

失敗予測として, XGBoost [3], L1 Logistic Regression, Random Forest の 3 つを使用した. これらのモデルは表形式データに対して高い予測性能を示すことが知られている.

a） 予測タスクの定義

本研究では, **ターン 3 までの対話情報を用いて, その後の推薦が失敗するかどうかを予測する二値分類タスク**として問題を定義した. 実用的な会話型推薦システムにおいて, 早期の失敗予測は, システムが適切な介入 (追加質問, 推薦戦略の変更など) を行うための十分な時間を確保できるため重要である.

b） ハイパーパラメータ設定

XGBoost モデルの学習には, 以下のハイパーパラメータを使用した:

- **max_depth**: 6 (決定木の最大深度)
- **learning_rate**: 0.1 (学習率)
- **n_estimators**: 100 (決定木の数)
- **scale_pos_weight**: 0.19 (クラス不均衡への対応)

その他のモデルについても, 同様にクラス不均衡を考慮したパラメータ設定を行った.

4.2.3 評価指標

モデルの性能評価には, 以下の指標を使用した:

- **AUC-ROC**: クラス識別能力の総合的な評価
- **Accuracy**: 全体的な予測精度
- **Precision**: 失敗と予測したもののうち実際に失敗した割合
- **Recall**: 実際の失敗をどれだけ検出できたか
- **F1-score**: Precision と Recall の調和平均

また, 特徴量の重要度分析を行い, 各特徴量が予測にどの程度寄与しているかを評価した.

4.2.4 統計的検定

モデル間の性能差の統計的有意性を評価するため, 以下の検定を実施した:

a） Accuracy の検定

Accuracy については, 予測結果のビット行列を用いて, Randomized Tukey HSD 検定 [2] を実施した. 有意水準は $p < 0.05$ とした.

b） AUC-ROC の検定

AUC-ROC については, ブートストラップ法による統計的検定を実施した. ブートストラップ反復回数は $n = 2000$ とし, 12 通りのモデル間比較 (4 つの特徴量セット ×3 つのモデル) を考慮して Holm 法による多重比較補正を適用した (有意水準 5%).

4.2.5 実験手順

実験は以下の手順で実施した:

1. データセットを訓練セット (80%) とテストセット (20%) に分割
2. 各特徴量セット (Baseline 1, Baseline 2, RQ1, RQ2) ごとに独立してモデルを学習
3. テストセットを用いてモデルの性能を評価
4. 特徴量の重要度を分析 (XGBoost)

4.3 実験結果

4.3.1 特徴量別の性能比較

表 3 に, 異なる特徴量セットを用いた場合の性能比較を示す. ベースライン手法 2 種類 (Baseline 1: 感情極性ベース, Baseline 2: 推薦結果の一貫性ベース) と提案手法 2 種類 (RQ1: 感情極性と変化度のミスマッチ, RQ2: 具体性と多様性のミスマッチ) について, XGBoost, L1 Logistic Regression, Random Forest の 3 つのモデルで評価を行った. 各提案手法とベースライン手法との間で AUC-ROC に関するブートストラップ検定 (2000 回リサンプリング) を実施し, Holm 法により多重比較補正を行った (12 回の比較, 有意水準 5%).

実験の結果, 提案手法はベースライン手法と比較して一貫して高い性能を示した. 特に, 提案手法 RQ1(感情極性と変化度のミスマッチ) を XGBoost で用いた場合に最も高い AUC-ROC(0.6355) を達成し, Baseline 1 と比較して約 22.5%, Baseline 2 と比較して約 19.4%の改善が見られた.

統計的検定の結果, AUC-ROC において以下の有意な改善が確認された:

提案手法 RQ1 の有効性: RQ1 は XGBoost および Random Forest において, Baseline 1 に対して統計的に有意な改善を示した ($p < 0.0042$). XGBoost では Baseline 2 に対しても有意な改善が確認された. 特に, XGBoost では Baseline 1 に対して+0.1169, Baseline 2 に対して+0.1032 の AUC-ROC 向上が得られた.

提案手法 RQ2 の有効性: RQ2 は XGBoost および Random

表 3 モデル別の評価指標比較

Model	AUC-ROC	Accuracy	Precision	Recall	F1-score
Baseline 1: 感情極性ベース					
XGBoost	0.5186	0.6214	0.86	0.68	0.76
L1 Logistic Regression	0.5676	0.5071	0.89	0.48	0.63
Random Forest	0.4669	0.7357	0.85	0.84	0.84
Baseline 2: 推薦結果の一貫性ベース					
XGBoost	0.5323	0.6286	0.88	0.66	0.75
L1 Logistic Regression	0.4956	0.5607	0.86	0.59	0.70
Random Forest	0.5243	0.7750	0.87	0.87	0.87
Proposed: RQ1 (感情極性と変化度のミスマッチ)					
XGBoost	**0.6355**^{*1,2}	0.7000	**0.89**	0.74	0.81
L1 Logistic Regression	0.5470	0.5500	0.87	0.56	0.68
Random Forest	0.5748^{*1}	0.7482	0.87	0.84	0.85
Proposed: RQ2 (具体性と多様性のミスマッチ)					
XGBoost	0.5961^{*1,2}	0.7000	0.87	0.77	0.81
L1 Logistic Regression	0.5675^{*2}	0.5339	0.88	0.53	0.66
Random Forest	0.6141^{*1,2}	**0.8089**	0.86	**0.93**	**0.89**

太字は全特徴量セットを通じた各評価指標の最高値を示す.
*はブートストラップ検定で Holm 法補正後に有意 (12 比較, 有意水準 5%).
上付き数字: [1] Baseline 1 との比較, [2] Baseline 2 との比較.

Forest において, Baseline 1 に対して統計的に有意な改善を示した. また, L1 Logistic Regression および Random Forest では Baseline 2 に対しても有意な改善が見られた. Random Forest での Baseline 1 に対する改善幅は+0.1472 と最も大きかった.

L1 Logistic Regression の傾向: L1 Logistic Regression では, RQ2 の Baseline 2 に対する比較でのみ統計的有意差が見られた. これは, L1 正則化による特徴選択が提案特徴量の効果を十分に活用できていない可能性を示唆している.

一方, Accuracy に関する Randomized Tukey HSD 検定では, いずれの比較においても統計的に有意な差は確認されなかった ($p > 0.05$). これは, クラス不均衡なデータセットにおいて, Accuracy がモデルの性能差を適切に捉えられていない可能性を示している. 対照的に, AUC-ROC はクラス不均衡に対してロバストな指標であり, 提案手法の優位性をより明確に示すことができた.

4.3.2 提案手法とベースライン手法による予測傾向の差異分析

ベースライン手法と提案手法の間で, 具体的に予測結果がどのように変化したかを分析した. ここでは, ベースライン手法が予測に失敗したデータ（誤判定）のうち, 提案手法によって正解へと改善された割合（改善率）と, 逆にベースライン手法での正解が不正解へと変化した割合（改悪率）を評価する.

図 1 より, 以下の傾向が確認された.

RQ2（具体性と多様性のミスマッチ）の補完的性質: RQ2 を導入した際, 多くのモデルにおいて改善率が高く, かつ改悪率が極めて低い（10%以下）傾向が見られた. これは, RQ2 が提供する情報が, 既存の感情極性や一貫性ベースの指標とは「独立した失敗パターン」を捉えていることを示唆している. すなわち, RQ2 は既存の判定ロジックを破壊することなく, 従来の指標では検知困難であった失敗事例をピンポイントで補完する性質を持つ.

RQ1（感情極性と変化度のミスマッチ）の予測方針の転換性: RQ1 を導入したケースでは, 改善率が非常に高い一方で, 改悪率も増大する傾向（図中の右側への移動）が確認された. これは, RQ1 が持つ情報が既存指標の予測根拠と強く干渉し, モデルの判断を大幅に書き換える性質を持つことを表している. この「予測の入れ替わり」の大きさは, RQ1 が既存指標とは全く異なる文脈（対話の動的なズレ）を評価していることを示している.

以上の分析から, RQ1 と RQ2 はそれぞれ異なるメカニズムで失敗予測に寄与しており, これらを既存指標と組み合わせることの妥当性が示された.

4.3.3 全特徴量を統合した時の性能比較

表 4 に, 全ての特徴量 (Baseline 1, Baseline 2, RQ1, RQ2) を統合したモデルの性能を示す. ベースライン手法と提案手法を組み合わせることで, 各手法を単独で用いた場合と比較してどの程度性能が向上するかを評価した.

全特徴量を統合したモデルは, いずれの手法を単独で用いた場合と比較しても高い性能を示した. 特に, XGBoost の AUC-ROC は 0.6531 となり, 提案手法 RQ1 単独 (0.6355) と比較して 2.8%の改善が見られた. これは, ベースライン特徴量と提案特徴量が相補的な情報を提供し, 統合することでより高精度な

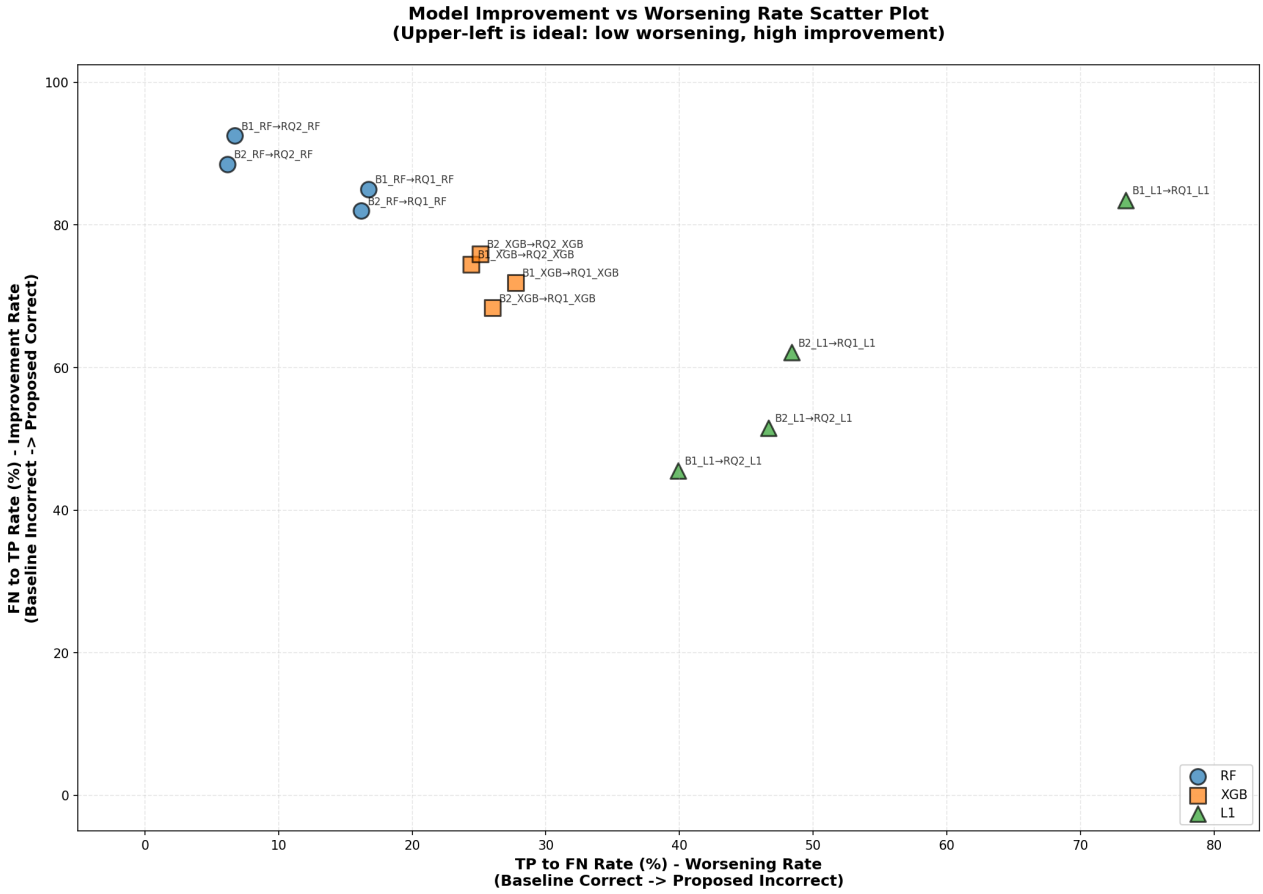

図 1　ベースライン手法から提案手法への予測変化の分析. 縦軸はベースラインの誤判定（FN）を提案手法が正解（TP）へ転換した割合を, 横軸はベースラインの正解（TP）が提案手法で誤判定（FN）へと変化した割合を示す.

表 4　全特徴量統合モデルの評価（太字は各評価指標における最高値を示す）

Model	AUC-ROC	Accuracy	Precision	Recall	F1-score
XGBoost	**0.6531**	0.7179	**0.89**	0.77	0.82
L1 Logistic Regression	0.6141	0.7750	0.87	0.87	0.87
Random Forest	0.6403	**0.8125**	0.88	**0.91**	**0.89**

失敗予測が可能になることを示している.

4.3.4　特徴量重要度分析

表 5 に, 全特徴量統合モデルにおける XGBoost の特徴量重要度を示す. これにより, 失敗予測においてどの特徴量が最も寄与しているかを定量的に評価できる.

分析の結果, 以下の知見が得られた:

提案特徴量の有効性:　上位 2 つの特徴量はいずれも提案特徴量 (RQ2 のターン 3 が 0.1389, RQ1 のターン 3 が 0.1244) であり, 提案手法の有効性が確認された.

ターン 3 の重要性:　上位 6 つの特徴量のうち, ターン 3 に関する特徴量が 5 つを占めており, 対話後半における情報が失敗予測に特に重要であることが示された.

表 5　XGBoost における特徴量の重要度

特徴量	重要度
RQ2: ターン 3 の具体性と多様性のミスマッチ	0.1389
RQ1: ターン 3 の感情極性と変化度のミスマッチ	0.1244
ターン 3 の感情極性 (Baseline 1)	0.1041
ターン 1 の感情極性 (Baseline 1)	0.0988
ターン 2 の感情極性 (Baseline 1)	0.0946
RQ2: ターン 2 の具体性と多様性のミスマッチ	0.0933
ターン 2 の推薦一貫性 (Baseline 2)	0.0916
RQ1: ターン 2 の感情極性と変化度のミスマッチ	0.0885
ターン 1 の推薦一貫性 (Baseline 2)	0.0843
ターン 3 の推薦一貫性 (Baseline 2)	0.0817

5　考　　察

5.1　提案手法の有効性

実験結果から, 以下の知見が得られた:

統計的に有意な性能向上: 統計的検定の結果, 提案手法は複数のモデルとベースライン手法の組み合わせにおいて, AUC-ROC の有意な改善を達成した. 特に, 提案手法 RQ1 は XGBoost において Baseline 1 に対して+0.1169 ($p \approx 0.0001$), Baseline 2 に対して+0.1032 ($p \approx 0.0002$) の改善を示し, Random Forest においても Baseline 1 に対して+0.1079 ($p \approx 0.0001$) の有意な改善が確認された. 同様に, 提案手法 RQ2 も XGBoost と Random Forest において両ベースライン手法に対して統計的に有意な改善を示した. これらの結果は, 提案手法の優位性が統計的に裏付けられたことを示している.

AUC-ROC による総合評価: 提案手法 RQ1(感情極性と変化度のミスマッチ) が XGBoost モデルにおいて最も高い AUC-ROC(0.6355) を達成した. この結果は, 単純な感情極性や推薦一貫性よりも, ユーザの感情変化とシステムの応答変化のミスマッチが失敗予測において重要な指標となることを示唆している. 特に, Random Forest を用いた RQ2 では, Baseline 1 に対して+0.1472 という最も大きな改善幅を達成しており, 具体性と多様性のミスマッチも強力な予測指標であることが確認された.

モデル依存性の観察: 統計的検定の結果, L1 Logistic Regression では提案手法による改善が他のモデルと比較して限定的であり, RQ2 の Baseline 2 に対する比較でのみ有意差が確認された ($p = 0.0023$). これは, L1 正則化による特徴選択が提案特徴量の効果を十分に活用できていない可能性を示唆している. 一方, XGBoost と Random Forest では, 提案手法が一貫して有意な改善を示しており, 決定木ベースのモデルが提案特徴量の非線形な関係性を効果的に捉えられることが示された.

Accuracy における有意差の不在: Randomized Tukey HSD 検定の結果, Accuracy においてはいずれの比較においても統計的に有意な差は確認されなかった. これは, クラス不均衡なデータセット (成功: 失敗 = 1:5) において, Accuracy がモデルの性能差を適切に捉えられていないことを示している. 多数派クラス (失敗ケース) の予測が全体的な精度を支配するため, 少数派クラス (成功ケース) の識別性能の違いが反映されにくい. 対照的に, AUC-ROC はクラス不均衡に対してロバストな指標であり, 提案手法の優位性をより明確に示すことができた. この結果は, クラス不均衡なデータセットにおける評価指標の選択の重要性を強調している.

特徴量統合による性能向上: 全特徴量を統合したモデル (表 4) は, 個別の手法と比較して一貫して高い性能を示した. XGBoost では, 提案手法 RQ1 単独 (AUC-ROC: 0.6355) から全特徴量統合 (AUC-ROC: 0.6531) へと 2.8%の改善が見られた. これは, 異なる観点からの特徴量を組み合わせることで, より包括的な失敗予測が可能になることを示している.

特徴量重要度に基づく有効性の検証: 表 5 に示す XGBoost の特徴量重要度分析は, 提案手法の有効性を定量的に裏付けている. 上位 2 つの特徴量はいずれも提案特徴量 (RQ2 のターン 3: 0.1389, RQ1 のターン 3: 0.1244) であり, 提案特徴量が上位 6 つのうち 3 つを占めた. 特に注目すべきは, ターン 3 に関する特徴量が上位 6 つのうち 5 つを占めている点である. これは, 対話後半における情報が失敗予測に特に重要であることを示しており, システムが対話の進行に伴って蓄積される情報を活用することで, より精度の高い予測が可能になることを示唆している.

また, ベースライン特徴量 (感情極性, 推薦一貫性) も上位 10 位以内に含まれていることから, これらの従来指標も提案特徴量と相補的な情報を提供していることが確認された. このことは, 提案手法が既存の指標を完全に置き換えるのではなく, それらと組み合わせることで効果を発揮することを示している.

5.2 ベースライン手法との比較

Baseline 1(感情極性ベース) は, いずれのモデルにおいても比較的低い AUC-ROC スコアを示した. 統計的検定においても, 提案手法との間に明確な有意差が確認されており, 感情極性単独では失敗予測に十分な情報を提供できないことが統計的に裏付けられた. 特に, ポジティブな感情を示しながらも推薦が失敗するケースや, ネガティブな感情を示しながらも最終的に成功するケースを適切に識別できない可能性がある.

Baseline 2(推薦結果の一貫性ベース) も, 提案手法と比較して統計的に有意に低い性能を示した. これは, 推薦結果の一貫性のみでは, ユーザの満足度や対話の成功を十分に予測できないことを示唆している. ただし, Baseline 2 は Baseline 1 と比較すると相対的に高い性能を示しており, 推薦の一貫性も一定の予測力を持つことが示された.

5.3 AUC-ROC の絶対値に関する考察

本研究における最高 AUC-ROC は, 全特徴量統合モデルの XGBoost で 0.6531 であった. この値はランダム予測 (0.5) を上回っており, 提案特徴量が失敗予測に有意義な情報を提供していることを示しているが, 実用的な観点からは更なる改善の余地がある.

AUC-ROC の絶対値が中程度にとどまっている要因として, 以下の点が考えられる. 第一に, Target-free な会話型推薦という設定そのものの予測困難性がある. ユーザが明確な目標を持たずに対話を開始し, 好みを動的に形成するため, 推薦の成否は対話の進行に伴って変化し, 早期段階での予測は本質的に困難である. 第二に, 本研究ではターン 3 までという限られた対話情報のみを用いている. 早期予測は実用上重要であるが, 対話初期段階で利用可能な情報量には限界がある. 第三に, 本研究で用いた特徴量は, 感情極性, 推薦一貫性, ミスマッチといった比較的少数の手設計特徴量に限定されており, 対話の内容を直接的にモデル化する深層学習ベースの手法は採用していない.

一方で, 本研究の主たる貢献は, AUC-ROC の絶対値を最大化することではなく, ユーザの発話特徴と推薦結果特徴の相互

作用が失敗予測に寄与するかという Research Question に答えることにある. ベースライン手法との比較において統計的に有意な改善が確認されたことは, ミスマッチ特徴量の有効性を実証するものであり, 今後, 対話内容の意味的表現や深層学習モデルとの統合によって, 絶対的な予測性能の向上が期待される.

5.4 本研究の成果

本研究の結果から, 会話型推薦システムの実装において以下のことが言える:

早期失敗検出: ターン 3 時点で統計的に有意な失敗予測が可能であることが示された. これにより, システムは推薦戦略を動的に調整したり, 追加の質問を行ったりすることで, 失敗を回避できる可能性がある.

多面的評価の重要性: 感情極性だけでなく, 具体性や多様性といった多面的な指標を組み合わせることで, より正確な失敗予測が可能となる. 統計的検定により, これらの指標の有効性が確認された.

ミスマッチの検出: ユーザの期待とシステムの応答のミスマッチを検出することが, 失敗予測において重要な役割を果たすことが統計的に示された.

評価指標の選択: クラス不均衡なデータセットにおいては, Accuracy よりも AUC-ROC のような不均衡に対してロバストな指標を用いることが重要であることが実証された.

6 研究の限界と今後の課題

本研究にはいくつかの限界があり, 今後の研究で対処すべき課題が残されている.

6.1 研究の限界

データセットのサイズと多様性: 本研究で使用したデータセットは 2,417 件の会話データから構成されているが, より大規模で多様なデータセットでの検証が必要である. 特に, 異なるペルソナや対話スタイルを持つユーザの会話データを増やすことで, モデルの汎化性能を向上させることができると考えられる. また, 統計的検定力をさらに高めるためにも, より大規模なデータセットでの追加検証が望ましい.

データの不均衡: 本研究で用いたデータセットには, 推薦の成功データと失敗データの間に不均衡 (約 1:5) が存在する. このようなクラス不均衡は, 機械学習モデルが多数派クラスに偏る傾向を生み, 少数派クラス (成功ケース) の予測精度に影響を与える可能性がある. 本研究では評価指標として AUC-ROC といった不均衡に対してロバストな指標を用いることで対処したが, より均衡の取れたデータセットでの検証が望ましい. 実際, Accuracy において統計的有意差が見られなかったことは, この不均衡の影響を示唆している.

ドメイン依存性: 本研究は映画推薦ドメインに限定されているため, 他のドメイン (音楽, 書籍, レストランなど) への汎化性能の検証が必要である. ドメインによって, ユーザの発話パターンや推薦の性質が異なる可能性がある.

シミュレータの限界: Pepper ユーザシミュレータを用いてデータを生成したため, 実際の人間のユーザとの対話とは異なる特性を持つ可能性がある. 実ユーザとの対話データを用いた検証が望ましい.

6.2 今後の課題

今後の研究では, 以下の方向性が考えられる:

適応的推薦手法の実装: 本研究で構築した失敗予測モデルを活用し, 予測結果に基づいてシステムが動的に推薦戦略を変更する仕組みの開発が期待される. 例えば, 失敗が予測される場合には, より詳細な質問を行ったり, 推薦の説明を強化したりすることが考えられる. 統計的に有意な予測が可能であることが示されたため, このようなシステムの実装がより現実的となった.

クラス不均衡への対処: データの不均衡問題に対処するため, SMOTE(Synthetic Minority Over-sampling Technique) などのオーバーサンプリング手法や, コスト考慮型学習 (cost-sensitive learning) の適用が考えられる. また, 今回サンプル数が少なかった成功ケースを意図的に収集するデータ収集戦略の設計も重要である. これにより, 少数派クラスの予測精度をさらに向上させ, Accuracy においても統計的に有意な差を検出できる可能性がある.

説明可能性の向上: 予測モデルがなぜ失敗を予測したのかを説明する機能を追加することで, システム開発者やユーザに対してより透明性の高いシステムを構築できる.

マルチモーダル情報の活用: テキスト情報だけでなく, 音声の韻律情報や対話のタイミングといったマルチモーダルな情報を組み込むことで, より高精度な失敗予測が可能になると考えられる.

実環境での評価: 実際の会話型推薦システムに本手法を組み込み, 実ユーザを対象とした評価実験を行うことで, 提案手法の実用性を検証する必要がある. 統計的に有意な性能向上が確認されたことから, 実環境での有効性が期待される.

7 結　　論

本研究では, Target-free な会話型推薦システムにおける推薦失敗予測の課題に取り組み, ユーザの発話特徴とシステムの推薦結果特徴の相互作用に着目した 2 つの新しいミスマッチ特徴量を提案した. 具体的には, 感情極性と推薦結果の変化度の関係を捉える特徴量 M_1(RQ1) と, ユーザ要求の具体性と推薦結果の多様性の関係を捉える特徴量 M_2(RQ2) を設計し, Pepper ユーザシミュレータによって生成された Target-free な会話データを用いてその有効性を検証した.

実験の結果, 提案手法はベースライン手法と比較して AUC-ROC で最大 22.5%の統計的に有意な改善を達成した. 特徴量重要度分析においても, 提案特徴量が上位 2 つを占め, その有効性が定量的に裏付けられた. また, 全特徴量を統合したモデルでは, 個別の手法を上回る性能が得られ, 提案特徴量と既存指標の相補的な関係が確認された.

これらの結果は, ユーザの期待とシステムの応答のミスマッチを検出することが, Target-free な会話型推薦における失敗予測に重要な役割を果たすことを示している. 本研究の成果は, 推薦の失敗が予測される場合に追加の質問や推薦戦略の変更といった介入を行う, より実用的な会話型推薦システムの構築に貢献するものである.

文　献

[1] Wanling Cai and Li Chen. Predicting user intents and satisfaction with dialogue-based conversational recommendations. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 33–42, 2020.

[2] Benjamin A Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Transactions on Information Systems (TOIS)*, Vol. 30, No. 1, pp. 1–34, 2012.

[3] Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.

[4] Sunghwan Kim, Kwangwook Seo, Tongyoung Kim, Jinyoung Yeo, and Dongha Lee. Stop playing the guessing game! target-free user simulation for evaluating conversational recommender systems. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

[5] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems 31 (NIPS 2018)*, 2018.

[6] Maria Vlachou. Failure prediction in conversational recommendation systems. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 599–604, 2025.

[7] Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. Fashion iq: A new dataset towards retrieving images by natural language feedback. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11307–11317, 2021.