

# 토스 NEXT ML Challenge CTR 예측: 실무진을 위한 연구 전략 가이드

## Executive Summary

본 문서는 토스 NEXT ML Challenge CTR 예측 대회(2025.09.08-10.13)에서 상위권 진입을 목표로 하는 AI 엔지니어를 위한 실무 중심 연구 전략을 제시합니다. 5주간의 예선 기간 동안 체계적인 접근을 통해 Private 리더보드 상위 30팀 진출과 본선 우승을 목표로 합니다.

**핵심 전략:** DCN-V3 기반 다단계 앙상블 + Average Precision 직접 최적화 + 적응적 불균형 처리

## 1. 대회 분석 및 전략적 접근

### 1.1 대회 특성 분석

데이터 규모 및 특성:

- 훈련 데이터: 10,704,179개 샘플, 119개 특성
- 테스트 데이터: 1,527,298개 샘플
- 익명화된 특성: 의미론적 해석 불가, 통계적 접근 필요
- 시간 특성: day\_of\_week, hour (시계열 패턴 존재)

평가 지표 전략:

- Average Precision (50%):** 순위 품질에 민감, 클래스 불균형 영향 큼
- Weighted LogLoss (50%):** 50:50 클래스 가중치로 균형 조정
- Public vs Private:** 30% vs 70% 분할로 과적합 위험 높음

경쟁 환경:

- 토스 실제 광고 데이터 기반으로 높은 실무 관련성
- 실시간 서빙 요구사항으로 모델 경량화 필요
- 본선 평가에서 코드 검증 및 보고서 평가 병행

### 1.2 승리 조건 분석

**1차 목표 (예선 통과):** Private 리더보드 상위 30팀

- 필요 추정 성능: Public 대비 +2-3% 상대적 향상
- 안정적 CV 전략으로 Public/Private 격차 최소화
- 15회 미만 제출로 Public LB 과적합 방지

**2차 목표 (본선 우승):**

- 모델 성능 (65점 만점): Private 점수 기반 상대 평가
  - 기술적 우수성 (35점): 혁신성, 해석가능성, 확장성
  - 재현 가능한 코드와 상세한 기술 보고서 필수
- 

## 2. 최신 SOTA 모델 연구 동향 (2023-2025)

### 2.1 DCN-V3: 게임 체인저 아키텍처

#### 핵심 혁신점:

- **Self-Mask Operation:** LayerNorm 기반 노이즈 필터링으로 파라미터 50% 감소
- **지수적 특성 상호작용:** 단 6개 레이어로 64차원 상호작용 포착 (기존 DCN-V2 대비 exponential growth)
- **Tri-BCE Loss:** 동적 손실 가중치로 하이퍼파라미터 튜닝 불필요

#### 성능 벤치마크:

- Criteo 데이터셋: 81.62% AUC (0.78M 파라미터)
- Avazu 데이터셋: 78.94% AUC
- KKBox 데이터셋: 84.31% AUC

#### 실무 적용 장점:

- 메모리 효율성: 기존 DCN 대비 50% 메모리 절약
- 추론 속도: sub-millisecond 지연시간 달성 가능
- 안정성: Tri-BCE로 학습 안정성 크게 향상

### 2.2 Transformer 기반 혁신 모델들

#### RAT (Retrieval-Augmented Transformer):

- Cross-sample 관계 모델링으로 cold-start 문제 해결
- Cascaded attention으로 intra/cross-sample 상호작용 동시 포착
- Long-tail 시나리오에서 기존 모델 대비 15-20% 성능 향상

#### AutoInt Enhancement:

- Multi-head self-attention with residual connections
- Feature embedding dimension adaptive scaling
- Attention weight visualization for interpretability

#### FT-Transformer for CTR:

- 수치형 특성을 위한 선형 임베딩 혁신

- CLS token을 통한 global context 학습
- 배치 정규화 대신 LayerNorm 사용으로 안정성 향상

## 2.3 경량화 및 배포 최적화 모델

### DistilledCTR Framework:

- Gated Ensemble Teacher로 다중 SOTA 모델 결합
- Knowledge distillation으로 20배 확장성 개선
- 5ms 미만 추론 지연시간 유지

### CAFE+ Compression (2024-2025):

- 10,000배 압축에도 최소 성능 손실
- HotSketch로 실시간 특성 중요도 추적
- Criteo에서 10,000배 압축시 오히려 +3.92% AUC 개선

---

## 3. 데이터 불균형 해결 전략

### 3.1 클릭률 불균형 특성 분석

#### 일반적 CTR 데이터 특성:

- 클릭률: 0.1-5% (극심한 불균형)
- Long-tail 분포: 소수 광고/지면이 대부분 클릭 집중
- 시간적 편향: 특정 시간대/요일에 클릭 집중
- 사용자 세그먼트별 큰 편차

#### 토스 데이터 특성 추정:

- 금융 앱 특성상 상대적으로 높은 클릭률 예상 (2-8%)
- B2C 금융 서비스의 계절성/시간성 패턴 강함
- 사용자 생애주기별 행동 패턴 차이 클 것으로 예상

### 3.2 고급 불균형 처리 기법

#### Focal Loss 최적화:

- Dynamic  $\alpha$  스케줄링: 학습 진행에 따라  $\alpha$ 값 조정
- Asymmetric Focal Loss: 양성/음성 클래스에 다른  $\gamma$ 값 적용
- Class-balanced Focal Loss: 클래스 빈도에 따른 자동 가중치

#### Advanced Sampling Strategies:

- **BorderlineSMOTE**: 경계선 근처 어려운 샘플 집중 생성
- **ADASYN**: 클래스 밀도에 따른 적응적 생성
- **Edited Nearest Neighbours**: 노이즈 샘플 제거 후 오버샘플링

#### **Cost-Sensitive Learning:**

- MetaCost: 비용 행렬 기반 메타 학습
- Threshold Moving: 최적 분류 임계값 동적 조정
- Ensemble Cost-Sensitive: 다양한 비용 함수로 앙상블

### **3.3 평가 지표별 특화 전략**

#### **Average Precision 최적화:**

- Ranking-aware loss functions
- Position-based weighting in training
- Precision@K optimization during validation

#### **Weighted LogLoss 최적화:**

- Balanced Cross-Entropy with pos\_weight=1.0
- Temperature scaling for probability calibration
- Mixup with label smoothing for regularization

---

## **4. 특성 공학 전략**

### **4.1 익명화 특성 처리 방법론**

#### **통계적 특성 분석:**

- Mutual Information으로 특성 중요도 측정
- Correlation matrix 기반 특성 클러스터링
- Distribution similarity를 통한 특성 타입 추론

#### **자동 특성 타입 감지:**

- Cardinality threshold 기반 범주형/수치형 분류
- Entropy 기반 정보량 측정
- Variance-to-mean ratio로 분포 특성 파악

#### **Domain-Agnostic Feature Engineering:**

- 빈도 기반 인코딩 (Frequency, Count encoding)

- 순위 기반 변환 (Rank transformation)
- 분위수 기반 이산화 (Quantile binning)

## 4.2 시간 특성 고급 활용

### 순환 시간 인코딩:

- Sine/Cosine 변환으로 주기성 보존
- Multi-scale temporal features (hour, day, week, month)
- Fourier transform 기반 주기 패턴 추출

### 시간 기반 집계 특성:

- Rolling statistics (1h, 6h, 24h, 7d windows)
- Exponential decay features with different half-lives
- Trend and seasonality decomposition

### 사용자 행동 패턴 모델링:

- 개인별 활동 시간 프로파일링
- 세션 기반 행동 시퀀스 분석
- Recency, Frequency, Monetary (RFM) 변형

## 4.3 High-Cardinality 특성 처리

### 해싱 기법:

- Feature hashing with collision monitoring
- Multiple hash functions for robustness
- Locality-sensitive hashing for similar items

### 임베딩 기반 접근:

- Entity embedding for categorical features
- Pre-trained embeddings from auxiliary tasks
- Hierarchical embedding for structured categories

### Target Encoding 고도화:

- K-fold target encoding with smoothing
  - Hierarchical target encoding
  - Noise injection for regularization
-

## 5. 모델 과적합 방지 전략

### 5.1 정규화 기법 조합

#### Multi-level Dropout:

- Input dropout for feature noise robustness
- Embedding dropout for representation learning
- Hidden layer dropout with scheduling
- Output dropout for prediction smoothing

#### Weight Regularization:

- L1/L2 regularization with adaptive coefficients
- Elastic Net combining L1 and L2
- Group Lasso for feature selection
- Spectral normalization for stable training

#### Data-level Regularization:

- Mixup for feature space interpolation
- CutMix for feature masking
- Label smoothing for confidence calibration
- Noise injection with curriculum learning

### 5.2 학습 안정화 기법

#### Gradient Optimization:

- Gradient clipping with adaptive norms
- Gradient accumulation for large batch simulation
- Gradient centralization for faster convergence
- SAM (Sharpness-Aware Minimization) for generalization

#### Learning Rate Scheduling:

- Cosine annealing with warm restarts
- Reduce on plateau with patience
- Cyclical learning rates with triangular policy
- One-cycle policy for fast convergence

#### Early Stopping Strategies:

- Multi-metric early stopping
- Patience scheduling based on validation curve
- Best model ensemble from multiple epochs
- Cross-validation based stopping criteria

## 5.3 Adversarial Validation

### Train-Test Distribution Shift Detection:

- Feature distribution comparison (KL divergence, JS divergence)
- Adversarial classifier for distribution matching
- Principal Component Analysis drift detection
- Statistical tests (KS test, Chi-square test)

### Robust Validation Strategy:

- Time-based splitting if temporal patterns exist
  - Stratified sampling with multiple criteria
  - Group K-fold for user-level splitting
  - Adversarial validation for feature selection
- 

## 6. 앙상블 전략 설계

### 6.1 모델 다양성 확보

#### Architecture Diversity:

- Tree-based models (LightGBM, XGBoost, CatBoost)
- Neural networks (DCN-V3, AutoInt, FiBiNET)
- Linear models (Logistic Regression, Linear SVM)
- Ensemble methods (Random Forest, Extra Trees)

#### Training Diversity:

- Different random seeds and initialization
- Various data preprocessing pipelines
- Different feature subsets and engineering
- Multiple cross-validation schemes

#### Hyperparameter Diversity:

- Bayesian optimization with different acquisition functions

- Grid search over different parameter spaces
- Random search with different distributions
- Evolutionary optimization for complex spaces

## 6.2 Stacking 최적화

### Multi-level Stacking:

- Level 1: Diverse base models with OOF predictions
- Level 2: Meta-models trained on level 1 outputs
- Level 3: Final ensemble with optimized weights
- Cross-validation for each level to prevent overfitting

### Meta-feature Engineering:

- Statistical summaries of base predictions
- Ranking and percentile features
- Prediction confidence and uncertainty measures
- Original feature interactions with predictions

### Meta-model Selection:

- Linear models for stable meta-learning
- Gradient boosting for non-linear combinations
- Neural networks for complex interactions
- Bayesian models for uncertainty quantification

## 6.3 블렌딩 최적화

### Weight Optimization Methods:

- Optuna for black-box optimization
- Scipy optimize for constrained optimization
- Genetic algorithms for global search
- Bayesian optimization for sample efficiency

### Dynamic Ensemble Strategies:

- Prediction-dependent weights
- Feature-dependent model selection
- Time-dependent weight adaptation
- Performance-based dynamic weighting



---

## 7. 실험 관리 및 최적화

### 7.1 체계적 실험 설계

#### **Ablation Study Framework:**

- Individual component contribution analysis
- Feature importance hierarchy establishment
- Hyperparameter sensitivity analysis
- Architecture component effectiveness study

#### **A/B Testing for ML:**

- Model comparison with statistical significance
- Online evaluation setup simulation
- Multi-armed bandit for hyperparameter tuning
- Bayesian testing for quick decisions

#### **Experiment Tracking:**

- Version control for data, code, and models
- Hyperparameter logging with automatic recording
- Metric tracking with visualization
- Reproducibility checks with seed management

### 7.2 하이퍼파라미터 최적화

#### **Bayesian Optimization:**

- Gaussian Process with acquisition functions
- Tree-structured Parzen Estimator (TPE)
- Multi-objective optimization (Pareto frontier)
- Early stopping integration for efficiency

#### **Population-based Methods:**

- Genetic algorithms for discrete spaces
- Particle swarm optimization for continuous spaces
- Differential evolution for complex landscapes
- Evolutionary strategies for neural architecture

## Gradient-based Methods:

- BOHB (Bayesian Optimization HyperBand)
- Population Based Training (PBT)
- Hypernetworks for continuous optimization
- Differentiable architecture search (DARTS)

## 7.3 모델 해석가능성

### Feature Importance Analysis:

- SHAP values for individual predictions
- LIME for local interpretability
- Permutation importance for global ranking
- Integrated gradients for neural networks

### Model Behavior Understanding:

- Partial dependence plots for feature effects
  - Interaction plots for feature combinations
  - Counterfactual explanations for decisions
  - Attention visualization for transformer models
- 

## 8. 5주 실행 계획

### 8.1 주차별 세부 일정

#### 1주차 (9/8-9/14): 기반 구축

- **Day 1-2:** 데이터 탐색 및 기본 EDA
  - 자동화 EDA 파이프라인 구축
  - 특성별 분포 및 클러스터 분석
  - 결측값 및 이상치 탐지
- **Day 3-4:** 기초 특성 공학
  - 시간 특성 엔지니어링
  - 범주형 특성 인코딩
  - 기본 상호작용 특성 생성
- **Day 5-7:** 베이스라인 모델 구축
  - 간단한 LightGBM/XGBoost 모델
  - 기본 DCN 모델 구현

- CV 전략 수립 및 검증

## 2주차 (9/15-9/21): 모델 개발

- **Day 1-3:** SOTA 모델 구현
  - DCN-V3 구현 및 튜닝
  - AutoInt, FiBiNET 모델 실험
  - RAT 모델 구현 (시간 허용시)
- **Day 4-5:** 고급 특성 공학
  - K-fold target encoding
  - High-cardinality 특성 처리
  - 도메인 특화 특성 생성
- **Day 6-7:** 불균형 처리 최적화
  - Focal Loss 파라미터 튜닝
  - 샘플링 전략 비교
  - Cost-sensitive learning 적용

## 3주차 (9/22-9/28): 앙상블 구축

- **Day 1-3:** 개별 모델 최적화
  - 하이퍼파라미터 자동 튜닝
  - 모델별 최적 특성 세트 탐색
  - OOF 예측 생성
- **Day 4-5:** Level-1 앙상블
  - 다양성 확보를 위한 모델 선택
  - 스택킹 메타 모델 훈련
  - 블렌딩 가중치 최적화
- **Day 6-7:** 앙상블 검증
  - CV 점수 vs LB 점수 상관관계 분석
  - Adversarial validation 수행
  - 과적합 징후 모니터링

## 4주차 (9/29-10/5): 고도화 및 최적화

- **Day 1-3:** Level-2 앙상블
  - 다단계 스택킹 구현
  - 메타 특성 엔지니어링
  - 앙상블 다양성 최적화

- **Day 4-5:** 성능 미세 조정
  - 예측 후처리 (calibration, smoothing)
  - 임계값 최적화
  - 평가 지표별 특화 조정
- **Day 6-7:** 최종 모델 선택
  - 안정성 vs 성능 트레이드오프 분석
  - Public LB 제출 전략 수립
  - 코드 정리 및 문서화 시작

## 5주차 (10/6-10/13): 최종 제출

- **Day 1-3:** 최종 검증
  - 재현성 확인 및 시드 고정
  - 에러 분석 및 마지막 개선
  - 앙상블 가중치 최종 조정
- **Day 4-5:** 제출 준비
  - 코드 정리 및 주석 작성
  - 기술 보고서 초안 작성
  - 제출 파일 검증 자동화
- **Day 6-7:** 전략적 제출
  - Public LB 과적합 방지
  - 최대 15회 제출 계획 수립
  - 최종 앙상블 제출

## 8.2 리스크 관리 계획

### Technical Risks:

- **메모리 부족:** 배치 크기 조정, 그래디언트 체크포인팅
- **과적합:** 강화된 정규화, 조기 종료
- **학습 불안정:** 학습률 스케줄링, 그래디언트 클리핑

### Strategic Risks:

- **Public LB 과적합:** 제한적 제출, CV 우선
- **코드 재현성 실패:** 버전 관리, 시드 고정
- **시간 부족:** 우선순위 기반 개발, 병렬 실험

### Contingency Plans:

- Plan A: DCN-V3 + Multi-level Stacking (최고 성능 목표)
  - Plan B: Proven SOTA models + Simple Blending (안정성 우선)
  - Plan C: LightGBM Ensemble + Feature Engineering (빠른 구현)
- 

## 9. 성공 지표 및 벤치마크

### 9.1 성능 목표 설정

정량적 목표:

- **Public LB:** 상위 10% 이내 유지
- **Private LB:** 상위 30팀 진입 (상위 약 15%)
- **CV vs LB 상관관계:** 0.85 이상 유지
- **양상블 향상도:** 단일 모델 대비 +3% 이상

정성적 목표:

- **코드 품질:** 재현 가능, 모듈화, 문서화
- **기술적 혁신:** 새로운 접근법 또는 기법 적용
- **실무 적용성:** 실시간 서빙 고려한 설계
- **해석가능성:** 비즈니스 인사이트 도출 가능

### 9.2 벤치마크 기준

내부 벤치마크:

- **Simple LightGBM:** 기본 특성만 사용
- **Advanced LightGBM:** 고급 특성 공학 적용
- **Single DCN-V2:** SOTA 단일 모델
- **Simple Ensemble:** 3-5개 모델 평균

외부 벤치마크:

- **Criteo Benchmark:** 공개 데이터셋 성능 비교
- **Industry Standards:** 실제 CTR 예측 시스템 성능
- **Academic SOTA:** 최신 논문 결과와 비교
- **Previous Competitions:** 유사 대회 우승 솔루션

### 9.3 실패 지표 및 대응

Early Warning Signals:

- CV와 LB 점수 divergence > 0.1
- 앙상블 성능이 단일 모델보다 낮음
- 메모리/시간 제약으로 실험 속도 저하
- 재현성 문제로 결과 불일치

#### 대응 전략:

- 성능 정체시: 특성 공학 강화, 새로운 모델 실험
- 과적합 징후시: 정규화 강화, 단순 모델로 회귀
- 시간 부족시: 우선순위 재조정, 병렬 처리 최대화
- 기술적 이슈시: 백업 플랜 활성화, 커뮤니티 도움 요청

---

## 10. 본선 진출 전략

### 10.1 코드 제출 준비

#### 코드 품질 기준:

- 재현성: 동일 환경에서 동일 결과 보장
- 모듈화: 기능별 명확한 분리, 재사용 가능
- 문서화: 상세한 주석, README, 실행 가이드
- 효율성: 메모리/시간 최적화, 불필요한 코드 제거

#### 제출 구조:

```
submission/  
├── README.md (실행 가이드)  
├── requirements.txt (의존성)  
├── config/ (설정 파일들)  
├── src/ (소스 코드)  
├── models/ (모델 가중치)  
├── notebooks/ (EDA, 실험 노트북)  
└── docs/ (기술 문서)
```

### 10.2 기술 보고서 작성

#### 필수 포함 내용:

- 문제 정의 및 접근 방법: 비즈니스 이해, 기술적 도전과제
- 데이터 분석 및 인사이트: EDA 결과, 도메인 특화 발견사항
- 모델링 전략: 아키텍처 선택 이유, 하이퍼파라미터 튜닝

- 실험 결과: Ablation study, 성능 비교, 오류 분석
- 앙상블 전략: 모델 조합 이유, 가중치 최적화 과정
- 비즈니스 가치: 실무 적용 방안, 예상 ROI, 확장성

#### 차별화 포인트:

- 혁신적 접근법: 새로운 아키텍처나 기법 도입
- 깊이 있는 분석: 단순 성능 보고를 넘어선 인사이트
- 실무 고려사항: 서버 최적화, A/B 테스트 설계
- 해석가능성: 모델 결정 과정 설명, 비즈니스 액션 제안

### 10.3 발표 준비 (시상식 대비)

#### 핵심 메시지:

- **Problem:** 왜 이 문제가 중요한가?
- **Solution:** 어떤 독창적 접근을 했는가?
- **Results:** 정량적 성과와 비즈니스 임팩트
- **Future:** 실무 적용 및 확장 가능성

#### 시각화 자료:

- 모델 아키텍처 다이어그램
- 성능 비교 차트 (ablation study)
- 특성 중요도 및 해석 결과
- 비즈니스 임팩트 시뮬레이션

---

## 결론

토스 NEXT ML Challenge에서 성공하기 위해서는 기술적 우수성과 전략적 접근이 모두 필요합니다. DCN-V3를 중심으로 한 SOTA 모델링, 체계적인 불균형 처리, 그리고 다단계 앙상블 전략을 통해 상위권 진입이 가능할 것입니다.

#### 성공의 핵심 요소:

1. 최신 연구 동향 반영: DCN-V3, RAT 등 2024-2025 SOTA 모델
2. 체계적 실험 관리: 재현 가능한 실험, 메트릭 추적, A/B 테스트
3. 도메인 특화 접근: 금융 앱 CTR의 특성을 반영한 특성 공학
4. 실무 지향적 설계: 실시간 서버, 해석가능성, 확장성 고려
5. 철저한 준비: 코드 품질, 기술 보고서, 발표 자료

이 가이드를 바탕으로 체계적으로 접근한다면, 토스 NEXT ML Challenge에서 우수한 성과를 거둘 수 있을 것입니다.