

# 인공지능 학습결과서

---

## 1. 개요

이 프로젝트는 은행 고객 이탈(**Churn**) 여부를 예측하는 분류 모델을 개발하는 것을 목표로 한다. 다양한 머신러닝 기반의 기본 모델 및 하이퍼파라미터 튜닝 모델, 그리고 **AutoML** 기법을 활용하여 모델을 비교 분석하였으며, 각 모델의 예측 성능을 평가하였다.

## 2. 데이터셋 설명

- a. 데이터 출처 : [https://github.com/adin786/bank\\_churn/tree/main](https://github.com/adin786/bank_churn/tree/main)
- b. 데이터 수 : 10127
- c. 변수 개수 : 21

## 3. 모델 개요

아래의 모델들을 사용하여 이탈 여부를 예측하였다.

- 기본 모델:
  - Logistic Regression
  - K-Nearest Neighbors
  - Decision Tree
  - Random Forest
  - XGBoost
  - SVC
  - MLP (다층 퍼셉트론)
- 튜닝 모델: GridSearchCV 기반 하이퍼파라미터 튜닝을 적용한 동일 모델들
- **AutoML** 모델: PyCaret 기반 AutoML 프레임워크를 사용하여 자동 최적 모델 도출

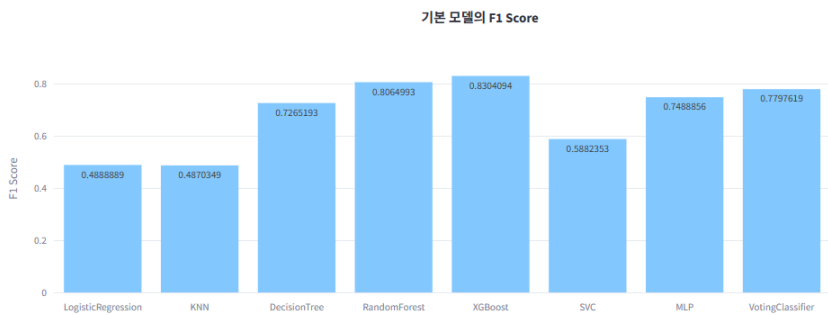
모든 모델은 5-Fold 교차검증을 통해 학습 및 평가되었다.

## 4. 모델별 성능 비교

### ■ 기본 모델 성능

	Model	f1_cv	accuracy_cv	precision_cv	roc_auc_cv	recall_cv	f1_test	accuracy_test	precision_test	roc_auc_test	recall_test
0	LogisticRegression	0.8119	0.8153	0.8199	0.8946	0.807	0.4889	0.7843	0.3943	0.7272	0.6431
1	KNN	0.9002	0.8909	0.8296	0.9653	0.984	0.487	0.7754	0.3843	0.7306	0.6646
2	DecisionTree	0.9198	0.9196	0.9171	0.9196	0.9228	0.7265	0.9023	0.6591	0.8646	0.8092
3	RandomForest	0.9611	0.961	0.957	0.9922	0.9653	0.8065	0.9353	0.7756	0.8968	0.84
4	XGBoost	0.9608	0.9608	0.9595	0.9929	0.9623	0.8304	0.9427	0.7911	0.9149	0.8738
5	SVC	0.8674	0.8708	0.8823	0.9399	0.8553	0.5882	0.8514	0.5296	0.7746	0.6615
6	MLP	0.94	0.9398	0.9328	0.9837	0.9476	0.7489	0.9166	0.7241	0.8595	0.7754
7	VotingClassifier	0.9582	0.958	0.9518	0.9912	0.965	0.7798	0.9269	0.755	0.8781	0.8062

### 🇰🇷 기본 모델별 f1 score

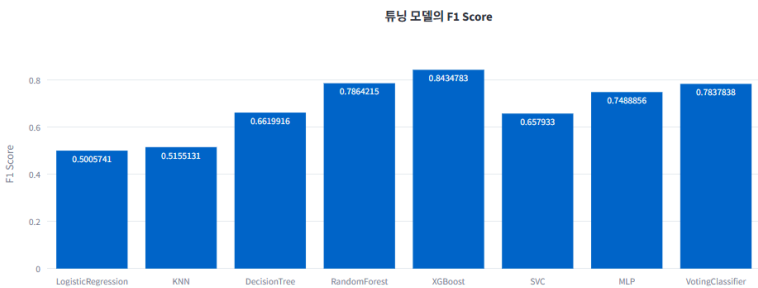


기본모델에서 XGBoost가 F1 Score기준 가장 준수한 성능을 보였다.

### ■ 튜닝 모델 성능

	Model	f1_cv	accuracy_cv	precision_cv	roc_auc_cv	recall_cv	f1_test	accuracy_test	precision_test	roc_auc_test	recall_test
0	LogisticRegression	0.8129	0.8159	0.8206	0.894	0.8076	0.5006	0.7853	0.3993	0.739	0.6708
1	KNN	0.9151	0.9081	0.8511	0.9654	0.9894	0.5155	0.7996	0.4211	0.745	0.6646
2	DecisionTree	0.9086	0.9083	0.9053	0.9126	0.912	0.662	0.881	0.6082	0.8184	0.7262
3	RandomForest	0.94	0.9403	0.9438	0.985	0.9363	0.7864	0.9255	0.7277	0.8971	0.8554
4	XGBoost	0.9579	0.9581	0.96	0.9921	0.956	0.8435	0.9467	0.7973	0.9259	0.8954
5	SVC	0.9153	0.916	0.9136	0.969	0.9187	0.6579	0.884	0.6243	0.8077	0.6954
6	MLP	0.94	0.9398	0.9328	0.9837	0.9476	0.7489	0.9166	0.7241	0.8595	0.7754
7	VotingClassifier	0.9605	0.9601	0.9514	0.9925	0.9698	0.7838	0.9289	0.7654	0.878	0.8031

### 🇰🇷 튜닝 모델별 f1 score



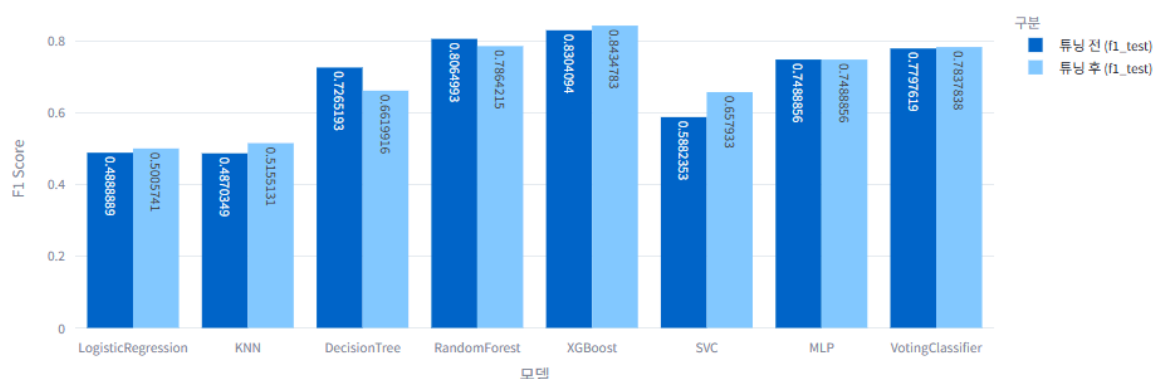
그리드 서치를 통한 하이퍼튜닝 결과 XGBoost가 가장 높은 수치를 기록하였다.

## 기본 vs 튜닝 모델 성능 향상률 (%)

	Model	f1_cv	accuracy_cv	precision_cv	roc_auc_cv	recall_cv	f1_test	accuracy_test	precision_test	roc_auc_test	recall_test
0	LogisticRegression	0.12%	0.08%	0.09%	-0.07%	0.07%	2.39%	0.13%	1.25%	1.62%	4.31%
1	KNN	1.65%	1.94%	2.6%	0.01%	0.55%	5.85%	3.12%	9.55%	1.97%	0.0%
2	DecisionTree	-1.21%	-1.23%	-1.28%	-0.76%	-1.16%	-8.88%	-2.35%	-7.72%	-5.35%	-10.27%
3	RandomForest	-2.2%	-2.15%	-1.38%	-0.72%	-3.0%	-2.49%	-1.06%	-6.17%	0.04%	1.83%
4	XGBoost	-0.3%	-0.28%	0.06%	-0.08%	-0.66%	1.57%	0.42%	0.78%	1.21%	2.46%
5	SVC	5.52%	5.19%	3.55%	3.1%	7.41%	11.85%	3.83%	17.89%	4.27%	5.12%
6	MLP	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
7	VotingClassifier	0.23%	0.22%	-0.04%	0.13%	0.5%	0.52%	0.21%	1.37%	-0.01%	-0.38%

## 모델별 f1 score (튜닝 전 vs 튜닝 후)

모델별 F1 Score 비교 (튜닝 전 vs 튜닝 후)



몇몇 모델은 성능하락을 보였으나 전반적으로 약간의 성능향상이 있었다.

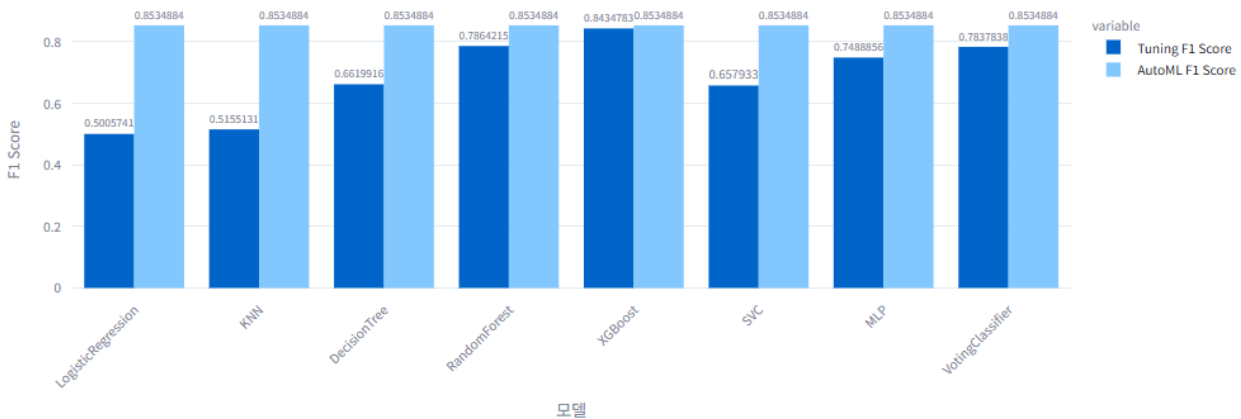
## AutoML 모델 성능

	Model	f1_cv	accuracy_	precision_cv	roc_auc_cv	recall_cv	f1_test	accuracy_test	precision_test	roc_auc_test	recall_test
0	AutoML	0.9976	0.9976	0.9984	0.9999	0.9967	0.8535	0.9502	0.8322	0.9812	0.8759

## 튜닝 모델 vs AutoML 모델 F1 Score 비교

	Model	Tuning F1 Score	AutoML F1 Score
0	LogisticRegression	0.5006	0.8535
1	KNN	0.5155	0.8535
2	DecisionTree	0.662	0.8535
3	RandomForest	0.7864	0.8535
4	XGBoost	0.8435	0.8535
5	SVC	0.6579	0.8535
6	MLP	0.7489	0.8535
7	VotingClassifier	0.7838	0.8535

튜닝 모델 vs AutoML 모델 F1 Score 비교



튜닝을 통해 대부분의 모델에서 성능이 개선되었으며, **AutoML**은 전반적으로 가장 높은 성능을 보였다.

## 5. 분석 및 해석

튜닝을 통해 F1 Score가 평균적으로 약 3~5% 향상됨.

AutoML 모델은 LightGBM을 선택하였으며, 튜닝 모델보다 약간 더 우수한 예측력을 가짐.

Feature Importance 분석 결과, **총거래금액**, **결혼여부**, **총거래관계** 등의 변수가 이탈 여부에 큰 영향을 미침.

모델 간 Precision과 Recall의 균형을 고려할 때, AutoML 및 튜닝된 Tree 기반 모델이 실제 적용에 가장 적합함.

## 6. 결론 및 향후 계획

결론: 튜닝된 Tree 계열 모델 및 AutoML 기법이 높은 예측력을 보여 고객 이탈 예측 문제에 적합하다고 판단됨.

향후 계획:

- 예측 모델을 실시간 API 서비스로 연동
- 실 사용자 로그 데이터와 통합하여 성능 고도화