세부 프로그램 명	기막두(기가_막힌_두뇌들)
창의역량그룹	

그림 4 우리 팀이 달성한 모델별 캐글 점수

구분	이름	학번	학년	소속대학	소속학과(부)
대표	조효원	20212241	3	창의 ICT 공과대학	전자전기공학부
팀원	나상현	20201876	3	소프트웨어 대학	소프트웨어학부
팀원	나영은	20203462	4	인문대학	역사학과
팀원	김소원	20221374	3	소프트웨어 대학	소프트웨어학부
팀원	박지후	20205338	4	경영경제대학	응용통계학과
팀원	정현석	20216010	2	창의 ICT 공과대학	전자전기공학부
학습주제	머신러닝 기초와 캐글 경진대회 참가				
팀 소개	참여동기 머신러닝은 의료, 자율 주행차, 로봇공학 등 다양한 분야에서 혁신을 이끌고 있습니다. 코드를 잘 다루는 것보다 데이터 전처리, 모델 선정 및 최적화 능력이 머신러닝에서는 필요합니다. 이러한 구축과정에서는 개개인의 경험과 창의성을 필요로하기에 서로 다른 학문 분야와 경험을 가진 팀원들끼리 모여 함께 학습하고 서로의 아이디어를 나눈다면, 더 나은 결과물을 얻을 수 있을 것이며 개인적인 성장뿐만 아니라 팀으로서의 역량도 향상시킬				

수 있을 것이라고 생각됩니다. 이러한 이유로 위 스터디에 참여하게 되었습니다.

이 스터디를 통해 다양한 프로젝트를 진행하고, 최종적으로 캐글 경진대회에 참여하여 유의미한 결과를 내도록 하였습니다.

팀원 소개와 각자의 목표

- 인공지능과 데이터 분석에 관심이 많은 6명이 모여 학습을 진행하였습니다.

그림 5 스터디 만남 후의 팀원과의 네컷

조효원

스마트 팩토리와 반도체의 품질 분석에 관심이 많아 데이터 분석 역량의 필요성을 느껴왔습니다. 데이터의 특성이나 이상치의 원인은 전공과목을 통해 습득할 수 있지만, 데이터를 다루는 방법을 배울 수는 없었습니다. 이 학습프로그램을 통해서 그 방법을 익혀나가고 싶었습니다.

나상현

생활에서 자주 쓰이는 ML 기법들의 수학적 기저와 그것을 사용하기 위한 데이터의 처리 등을 학습하며 향후 프로젝트에서 활용할 수 있는 기반을 닦고자 한다

나영은

수학통계적 원리를 활용해 머신러닝의 전체적인 개념을 파악하고

	캐글 예제를 통해 직접 실습해보면서 프로젝트에서 혼자 적용할 수 있는 능력을 기르고자 한다. 김소원 머신러닝 알고리즘의 개념과 적용 방법을 이해하고 실습 과제를 통해 적용 능력을 기르고자 한다. 스터디원과의 협업으로 심도 깊은 학습을 하는 것이 목표이다. 박지후 저는 생물통계 분야에 관심이 있으며, 이를 위해 데이터마이닝 전공 수업을 듣고 있습니다. 데이터마이닝 기술을 활용하여 생물학적 데이터를 분석하고 유의미한 패턴을 도출하는 방법을 배우고 있습니다. 이를 통해 질병 예측 및 유전자 분석과 같은 다양한 생물학적 문제를 해결하는 데 기여하고자 합니다. 머신러닝 알고리즘과 통계적 개념을 복습하고 다양한 실습을 통해 직접 활용해 보고자 합니다. 또한, 배웠던 알고리즘을 새로운 프로젝트에 활용해 볼 계획입니다. 정현석 항상 가볍게 머신러닝에 대해서 접했던터라 1차적으로 <파이썬 머신러닝 완벽 가이드>의 책을 따라가면서 차근차근 수학적인 내용과 함께 기초를 배우고 싶다. 특히 분류 모델 중 XGBoost, LightGBM 등과 같은 방법들 익히고 싶다. 이런 것을 통해서 최종적으로 책에 나와있는 캐글 예제들도 풀면서, 스스로 ML를 다룰 수 있는 수준이 되는 것이 이번 학기 스터디의 목표이다.

1. 활동 개요

활동 주제		머신러닝 기초와 캐글 경진대회
활동 목표		· 데이터 전처리, 모델 선정, 최적화 능력을 함양한다. · 캐글 경진대회에 참석하여 유의미한 결과를 얻는다. · 각 모델의 필요성에 대하여 각자의 전공의 관점에서 중요한 이유를 차례대로 격주씩 발표한다.
주요 자료		· 파이썬 머신러닝 완벽가이드(개정 2판) · kaggle · Dacon
일정		**주요 활동 내용**
1회	기초 내용 복습 (머신러닝 생태계, 사이킷런) 평가 및 분류	-머신러닝의 기본이 되는 파이썬의 기본 라이브러리(넘파이, 판다스)에 대하여 복습하고 코드를 실습함. -평가 및 분류에 대하여 책을 읽고 예제코드를 제출함. -기본 역량이 중요한 만큼 머신러닝의 기본에 대하여 zoom meeting을 진행 시에 퀴즈를 풀고 ppt를 제작하며 완전히 익힌다.
2회	분류 코드 복기	분류에 대한 코드를 리뷰함. 전공발표 1. 이미지 및 영상 신호처리에 사용되는 분류 모델 및 segmentation에 대하여
3회	분류 2	Light GBM과 XGBoost 등 분류의 심화 모델에 대하여 학습. 전공발표 2. LightGBM을 사용한 역사적 데이터 패턴 분석 논문 리뷰
4회	회귀	-회귀에 대하여 책을 읽고 예제 코드를 제출함. -줌 미팅 과정에서 회귀에 대한 퀴즈 풀이를 진행함.
5회	회귀 코드 복기	- 제출한 과제 코드를 분석한다. - 실제 데이터 set(반도체 박막 분석)에 대한 실습을 진행함 - 전공발표 3. 회귀를 이용한 반도체 박막 두께 분석(dacon 데이터 활용)
6회	차원축소	- 차원축소에 대하여 책을 읽고 예제 코드를 제출함 - 줌 미팅 과정에서 퀴즈 풀이를 진행함 - 차원축소에 대한 ppt 발표를 진행하며 학습을 강제시함.
7회	차원축소 코드 복기	- 제출한 과제 코드를 분석한다. - 전공발표 4. CV에 사용되는 PCA와 군집화 기술 - 전공발표 5. 6G eURLLC 통신 성능향상을 위한 DRL(강화학습) 논문 리뷰
8회	군집화 학습 및 경진대회 선정	- clustering에 대하여 학습하고 예제 코드를 제출함 - 퀴즈 풀이를 진행함 - 경진대회 주제를 선정함.

9회	최적의 EDA 선정 및 마무리	- 선정한 경진대회 주제인 'spaceship titanic'에 대하여 각자 EDA를 수행함 - 수행한 결과에 대하여 논의하고 최적의 EDA 기법을 융합함.

2. 활동 성찰

□ 운영 노하우

1. 대면 스터디를 통해 강제성을 부여한다.

2회차 당 1번은 꼭 대면 스터디를 통해 진행하였다. 이를 통해 과제 수행 여부를 점검하고, 실제 코드를 공유하며 강제성이 생겨 모든 팀원이 과제를 놓친적이 존재하지 않는다.

2. PPT 제작과 발표를 진행하며 스터디를 일깨운다.

책의 내용이 워낙 길고, 예제 코드도 많기에 학습 진행과정에서 대충 읽고, 과제를 익히지 않고, 복사 붙여넣기를 하며 의지가 약해질 수 있다. 그렇기 때문에 ppt 발표를 진행한다면 제작이나 발표를 위해서라도 열심히 공부할 것이다. 따라서 총 3회의 발표를 진행하였다.

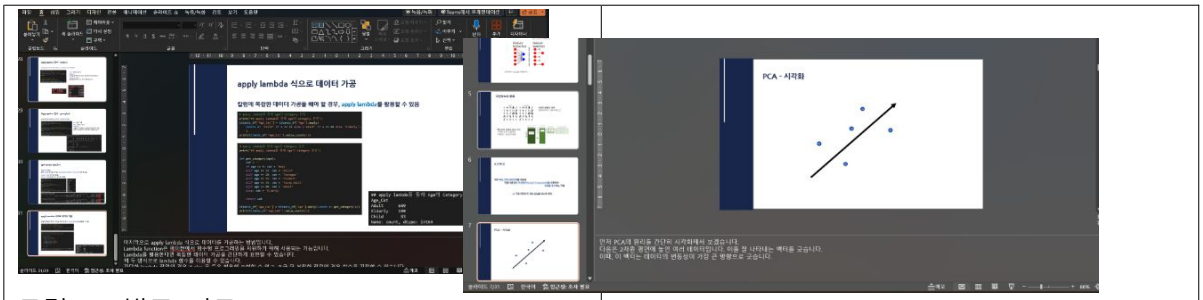


그림 10 발표 자료 PPT

3. 학습을 점검할 수 있는 수단을 만든다.

과제와 퀴즈를 통해서 학습을 점검했다. 퀴즈는 각자 만들어 온 후 zoom 스터디 과정에서 퀴즈를 합치고 풀어나갔다. 복습과 점검 수단을 만들고 그동안의 학습을 성찰하였다.

그림 12 과제를 제출하는 github에 해당

그림 13 나영은 7장 과제

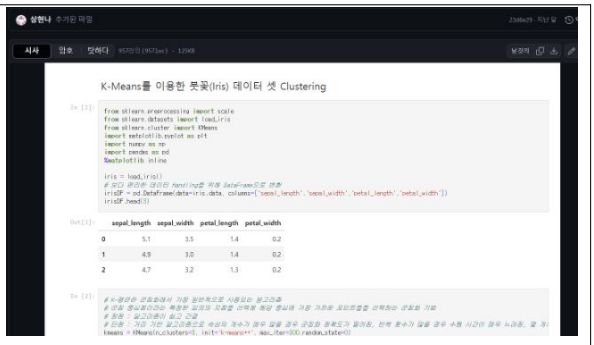


그림 14 나상현 7장

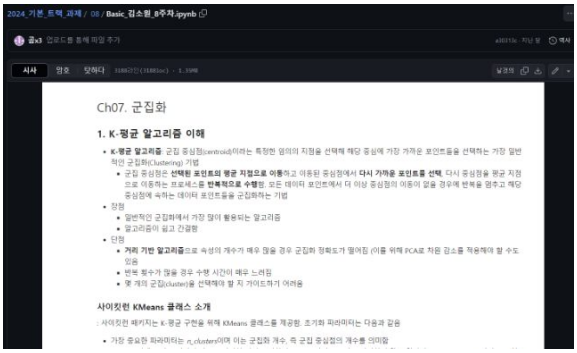


그림 15 김소원 7장 과제

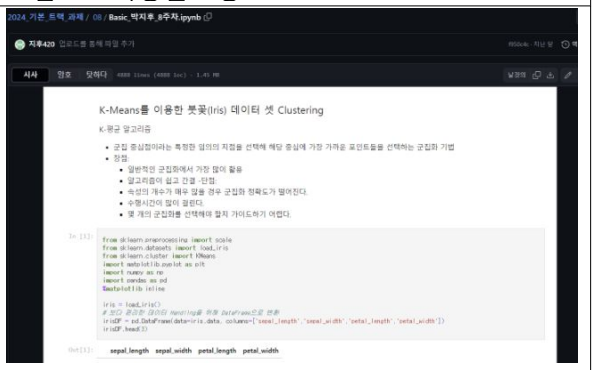


그림 16 박지후 7장 과제

그림 17 조효원 7장 과제

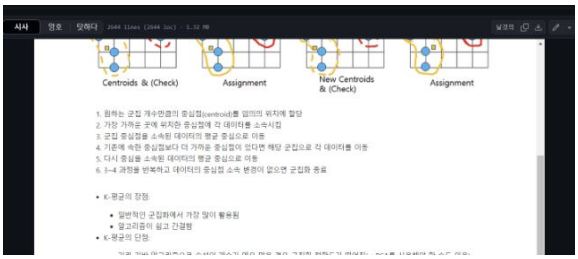


그림 18 정현석 7장 과제

□ 학습성과

1. kaggle 점수 0.8 달성

우리의 최종 목표는 캐글 경진대회에 참석하는 것이다. 예제 코드를 통해서 캐글을 많이 다루어봤지만, 책에 나온 내용을 그대로 읽고 옮겨적은 것이다. 따라서 우리가 직접 그 데이터의 특성을 이해하고 배운 내용을 적용하여 0.8이 넘는 높은 정확도의 예측하는 모델을 설계할 수 있었다. 경진대회를 참석하는 것이 우리 팀의 최종 목표였기 때문에 데이터 분석기법을 자세히 설명할 수 있는 ppt를 제작하였다. 이미지로 첨부한다.

캐글 경진대회 점수

적용된 모델

Random Forest
여러 결정 트리의 보팅으로 최종 결정하는 앙상블 알고리즘

LGBM
XGBoost와 함께 가장 각광받는 부스팅 알고리즘
학습시간 적고, 리프 중심의 트리 분할 방식을 사용함

+ K-fold CV
데이터를 k개의 폴드로 나누어 모델을 평가하는 방법
테스트 데이터를 활용하지 않고도 훈련성능을 평가할 수 있음

+ HyperOpt
베이지안 최적화 기법을 사용하여 최적의 하이퍼파라미터 조합을 찾는 라이브러리

Random Forest

sklearn.ensemble 의 RandomForestClassifier 활용

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

# X와 y 설정
X = df5.drop(columns=['Transported'])
y = df5['Transported']

# 학습 데이터와 테스트 데이터로 분할
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Random Forest 모델 학습
model1 = RandomForestClassifier(random_state=42)
model1.fit(X_train, y_train)

# 모델 예측
y_pred = model1.predict(X_test)

# 성능 평가
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Model Accuracy: {accuracy:.4f}")
print("Classification Report:")
print(report)
```

〈 Accuracy 〉

Train : 0.7798
Kaggle : 0.78933

.csv
Complete · now
0.78933

LightGBM + HyperOpt

lightgbm, hyperopt

```
# 하이퍼파라미터 공간 정의
space = {
    'objective': 'binary',
    'boosting_type': 'gbdt',
    'metric': 'binary_logloss',
    'num_leaves': scope.int(hp.quniform('num_leaves', 20, 40, 1)),
    'learning_rate': hp.uniform('learning_rate', 0.01, 0.2),
    'feature_fraction': hp.uniform('feature_fraction', 0.5, 1.0),
}

# 목적 함수 정의
def objective(hyperparams):
    model = lgb.train(
        hyperparams,
        d_train,
        num_boost_round=100,
        valid_sets=[d_test],
        valid_names=['valid'],
        callbacks=[lgb.early_stopping(stopping_rounds=10)],
        # verbose_eval=0 → 학습 과정을 출력하지 않음
    )
    y_pred_prob = model.predict(X_test, num_iteration=model.best_iteration)
    y_pred = np.round(y_pred_prob)
    accuracy = accuracy_score(y_test, y_pred)

    # 최적화할 목표를 위해, 이 경우 정확도의 음수 값을 반환하여 최대화 문제로 변환
    return {'loss': -accuracy, 'status': STATUS_OK}

# 최적화 실행
trials = Trials()
best = fmin(fn=objective,
            space=space,
            algo=tpe.suggest,
            max_evals=100,
            trials=trials)
```

〈 Accuracy 〉

Train : -
Kaggle : 0.80734

LightGBM.csv
Complete · 25m ago
0.80734

LightGBM + HyperOpt + K-Fold CV

lightgbm, hyperopt, StratifiedKFold

```
def objective(params):
    accuracies = []
    params['num_leaves'] = int(params['num_leaves'])
    skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

    for train_index, test_index in skf.split(X, y):
        X_train, X_test = X.iloc[train_index], X.iloc[test_index]
        y_train, y_test = y.iloc[train_index], y.iloc[test_index]

        # LightGBM 데이터셋 생성
        d_train = lgb.Dataset(X_train, label=y_train)
        d_test = lgb.Dataset(X_test, label=y_test)

        # 모델 학습
        evals_result = {}
        model = lgb.train(
            params,
            d_train,
            num_boost_round=100,
            valid_sets=[d_test],
            valid_names=['valid'],
            # verbose_eval=False
        )

        # 모델 예측
        y_pred_prob = model.predict(X_test, num_iteration=model.best_iteration)
        y_pred = np.round(y_pred_prob)

        # 정확도 계산
        accuracy = accuracy_score(y_test, y_pred)
        accuracies.append(accuracy)
```

〈 Accuracy 〉

Train : 0.8036
Kaggle : 0.80243

k-fold cvLightGBM .csv
Complete · 25m ago
0.80243

이 결과를 얻기 위한 노력들

데이터 분석 주제

● **대회 주제**

Spaceship Titanic : Predict which passengers are transported to an alternate dimension

● **선정 이유**

실종된 승객을 구출하기 위해, 우주선의 손상된 컴퓨터 시스템에서 복구된 기록을 사용하여 변칙 현상에 의해 이송된 승객을 예측

→ 상상력을 기반한 새로운 칼럼 생성 가능

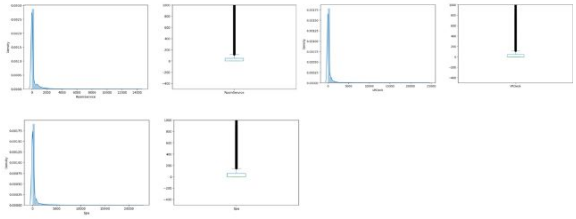
데이터 소개

X variable
- **PassengerId**: 각 승객의 고유 ID
- **HomePlanet**: 승객이 출발한 행성, 일반적으로 영구 거주 행성.
- **CryoSleep**: 냉동 수면 여부
- **Cabin**: 승객이 머무는 객실 번호
- **Destination**: 승객이 출발할 행성.
- **Age**: 승객의 나이
- **VIP**: 승객이 항해 중 특별 VIP 서비스에 대한 비용을 지불했는지 여부.
- **RoomService, FoodCourt, ShoppingMall, Spa, VRDeck**
  : 승객이 타이타닉 우주선의 다양한 고급 편의 시설에 대해 청구한 금액
- **Name**: 승객의 이름과 성.

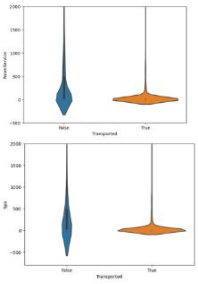
Target variable
- **Transported**: 승객이 다른 차원으로 이송되었는지 여부

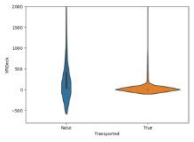
소비 칼럼 범주화

소비 칼럼 범주화



소비 칼럼 범주화



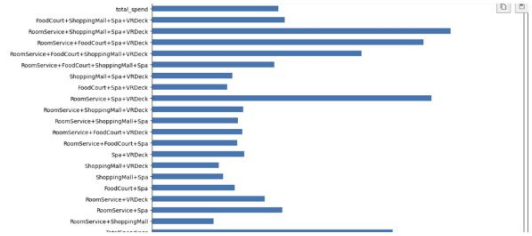


RoomService, Spa, VRDeck이
유사한 분포를 가진다.

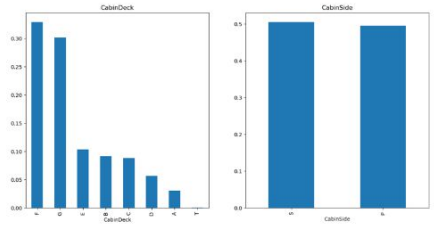
소비 칼럼 범주화

```
columns = ['RoomService', 'FoodCourt', 'ShoppingMall', 'Spa', 'VRDeck']

# 2개의 다항 합계
for combo in combinations(columns, 2):
    col_name = '+'.join(combo)
    df[col_name] = df[list(combo)].sum(axis=1)

# 3개의 다항 합계
for combo in combinations(columns, 3):
    col_name = '+'.join(combo)
    df[col_name] = df[list(combo)].sum(axis=1)

# 4개의 다항 합계
for combo in combinations(columns, 4):
    col_name = '+'.join(combo)
    df[col_name] = df[list(combo)].sum(axis=1)

# 5개의 다항 합계
for combo in combinations(columns, 5):
    col_name = '+'.join(combo)
    df[col_name] = df[list(combo)].sum(axis=1)

df.rename(columns={'RoomService+FoodCourt+ShoppingMall+Spa+VRDeck': 'total_spend'}, inplace=True)
df.head()
```

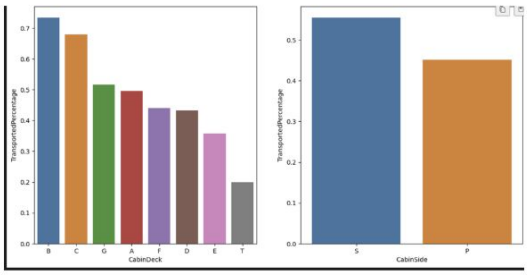
소비 칼럼 범주화



Cabin 칼럼 범주화



Cabin 칼럼 범주화



Cabin 칼럼 범주화

Cabin 칼럼

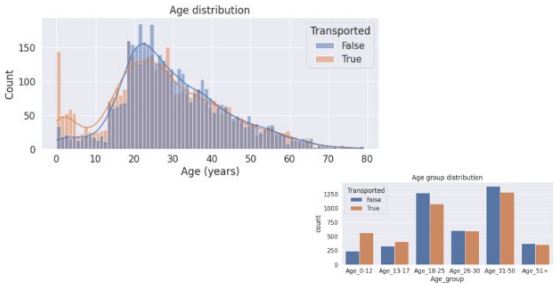
- 승객이 머무는 객실 번호
- deck/num/side

```
df[['Deck', 'Cabin_no', 'Side']]= df['Cabin'].str.split('/', expand= True)
df = df.drop('Cabin', axis=1)
df.head()
```

	PassengerId	HomePlanet	CryoSleep	Destination	Age	VIP	RoomService	FoodCourt	ShoppingMall	Spa	...	ShoppingMall+Spa+VRDeck	RoomServic
0	0001_01	Europa	False	TRAPPIST-1e	39.0	False	0.0	0.0	0.0	0.00	...	0.00	
1	0002_01	Earth	False	TRAPPIST-1e	24.0	False	109.0	9.0	25.0	549.00	...	618.00	
2	0003_01	Europa	False	TRAPPIST-1e	58.0	True	43.0	3576.0	0.0	6715.00	...	5743.52	
3	0003_02	Europa	False	TRAPPIST-1e	33.0	False	0.0	1283.0	371.0	3329.00	...	3893.00	
4	0004_01	Earth	False	TRAPPIST-1e	16.0	False	303.0	70.0	151.0	565.00	...	718.00	

5 rows × 44 columns

Age 칼럼 범주화



Age 칼럼 범주화

```
# Update age group feature
df.loc[df['Age']<=12,'Age_group']='Age_0-12'
df.loc[(df['Age']>12) & (df['Age']<18),'Age_group']='Age_13-17'
df.loc[(df['Age']>=18) & (df['Age']<=25),'Age_group']='Age_18-25'
df.loc[(df['Age']>25) & (df['Age']<=30),'Age_group']='Age_26-30'
df.loc[(df['Age']>30) & (df['Age']<=50),'Age_group']='Age_31-50'
df.loc[df['Age']>50,'Age_group']='Age_51+'
```

MemberCount 칼럼

- passenger id : 'gggg_dd'

```
df['Group']=df['PassengerId'].astype(str).str[:4].astype(int)
```

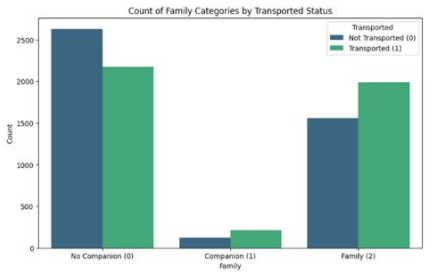
- Last Name

```
df['LastName'] = df['Name'].str.split().str[0]
```

- MemberCount

```
# 동승자의 수
df['MemberCount'] = df['Group'].map(df['Group'].value_counts())
# 동승자가 없으면 0, 있으면 1, 가족이면 2. 가족인지 판단은 LastName이
중복되는지 여부로 판단
df['Family'] = 0
df.loc[df['MemberCount'] > 1, 'Family'] = 1
df.loc[(df['MemberCount'] > 1) &
(df['LastName'].duplicated(keep=False)), 'Family'] = 2
```

MemberCount 칼럼

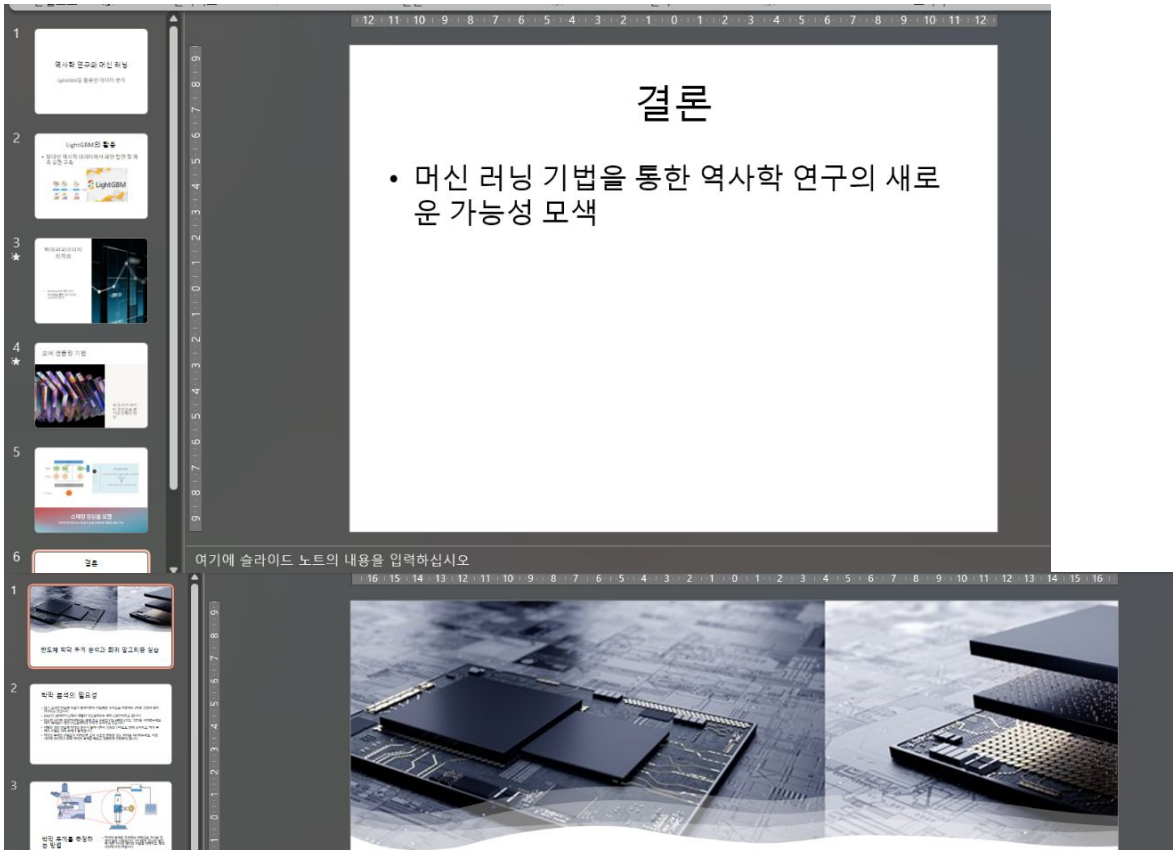


2. 파이썬 머신러닝 완벽 가이드 책 완독

책 한권을 끝내는 것은 쉽지 않다. 더불어 학술적인 도서 한권을 온전히 익히는 것은 더더욱 어렵다. 하지만 이 스터디를 통해 600p가 넘는 책 한권을 온전히 쌓을 수 있었다.

3. 전공관련 논문 리뷰 및 프로젝트 참가

각자의 전공에 관해서 머신러닝의 적용 분야에 대하여 프로젝트를 참가하거나 데이콘 등의 자료를 이용해서 코드를 설계해보고 그 내용을 zoom 미팅 시간에 발표하였다.



□ 우리 팀이 개선해야 할 사항

1. 약속 시간 지키기

아무래도 고학년들로 이루어지다 보니 학부연구생이나 팀 프로젝트가 많아서 약속을 잡기란 쉽지 않았다. 그래서 약속된 시간에 항상 변경이 있었다. 공통된 시간을 찾는 것이 너무 어려웠다.

2. 스터디 내용을 정리하는 방법이 필요!

스터디를 진행하였지만, 그 내용을 코드 외에 정리하지 않아 성과물이 한눈에 보이지 않는다. 그렇기 때문에 스터디 과정에서 배운 내용 익힌 내용을 블로그나 노트로 정리하여 다른 사람들에게도 도움이 되도록 학습물을 공유했다면 어떨까 성찰한다.

□ 우리 팀의 향후 계획

1. 다양한 데이터 분석 실전 대회 참전

이 스터디를 통해서 기본적인 데이터 전처리 및 모델 선정, 최적화 방법을 배웠으며 실전 경진대회에도 참석하여 높은 정확도를 얻었다. 더불어 대회에서 서로의 의견을 공유하고 다른 사람의 코드를 참고하여 하나의 결과물을 얻어가는 과정에서 많은 어려움을 동반한 성장을 느꼈다. 그렇기에 부족한 부분이 있더라도 실전 대회에 참가하여 부딪힌다면 유의미한 결과를 얻을 수 있다고 생각한다.