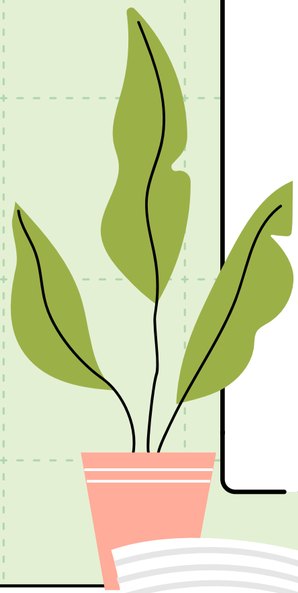
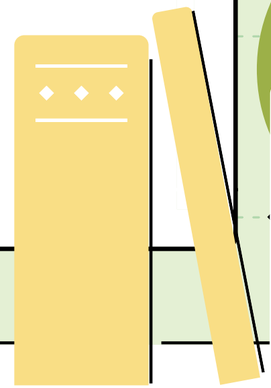
인공지능

인공신경망

이름 깡인환

학번 214683





건 남 대 학 교

Contents

모델링 배경 및 목적

여름이 다가오는 만큼 같이 다가오는 장마철, 장마철에 대비해 미리 비오는 날을 예측하는 모델을 만들어 보고자 Kaggle에서 날씨 예측 데이터셋을 가져왔습니다.

날씨는 다양한 변수 요인에 영향을 받기 때문에 단순한 규칙 예측이 아닌 인공신경망을 통한 다차원적 데이터를 활용하여 모델링 하고자 하였습니다.

데이터 설명

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Date           145460 non-null  object
 1   Location       145460 non-null  object
 2   MinTemp        143975 non-null  float64
 3   MaxTemp        144199 non-null  float64
 4   Rainfall       142199 non-null  float64
 5   Evaporation    82670 non-null   float64
 6   Sunshine       75625 non-null   float64
 7   WindGustDir    135134 non-null  object
 8   WindGustSpeed  135197 non-null  float64
 9   WindDir9am     134894 non-null  object
 10  WindDir3pm     141232 non-null  object
 11  WindSpeed9am   143693 non-null  float64
```

22개의 독립 변수, 1개의 종속변수가 존재(RainTomorrow)

변수 4가지를 제거하여 19개의 변수 사용

데이터 145,459개 존재

```
 12  WindSpeed3pm   142398 non-null  float64
 13  Humidity9am    142806 non-null  float64
 14  Humidity3pm    140953 non-null  float64
 15  Pressure9am    130395 non-null  float64
 16  Pressure3pm    130432 non-null  float64
 17  Cloud9am       89572 non-null   float64
 18  Cloud3pm       86102 non-null   float64
 19  Temp9am        143693 non-null  float64
 20  Temp3pm        141851 non-null  float64
 21  RainToday      142199 non-null  object
 22  RainTomorrow   142193 non-null  object
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

모델링 과정

라이브러리 설정 및 csv 파일 가져오기

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import roc_curve, auc, precision_recall_curve, average_precision
import matplotlib.pyplot as plt
import tensorflow as tf

# 데이터 불러오기
df = pd.read_csv("weatherAUS.csv")
df = df.drop(columns=["Evaporation", "Sunshine", "Cloud9am", "Cloud3pm"])
df = df.dropna(subset=["RainTomorrow"])
df[df.select_dtypes(include="float64").columns] = df.select_dtypes(include="float64")
```

불필요 속성 : "Evaporation", "Sunshine", "Cloud9am", "Cloud3pm"

⇒ 결측지가 많은 속성들, 목표값 또한 제외

수치형 데이터 → 평균으로 대체

범주형 데이터 → incoding

입력 및 출력 구분 및 정규화

```python
categorical_cols = df.select_dtypes(include=["object"]).columns.drop(["Date", "RainTomorrow"])
df = pd.get_dummies(df, columns=categorical_cols)
df["RainTomorrow"] = df["RainTomorrow"].map({"No": 0, "Yes": 1})
df = df.drop(columns=["Date"])

X = df.drop(columns=["RainTomorrow"]).values
y = df["RainTomorrow"].values

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_cnn = X_scaled.reshape((X_scaled.shape[0], X_scaled.shape[1], 1, 1))

X_train, X_test, y_train, y_test = train_test_split(X_cnn, y, test_size=0.2, random_state=42, stratify=y)
```

X : 기상데이터 (입력), Y 내일 비 (출력)

정규화 (평균0, 표준편차1), 스케일조정 (2차원)

학습 데이터 : 8, 테스트 데이터 : 2로 구분

모델링 과정

CNN 모델 학습

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(16, (3, 1), activation='relu', input_shape=(X_train.shape[1], 1, 1)),
    tf.keras.layers.MaxPooling2D(pool_size=(2, 1)),
    tf.keras.layers.Conv2D(32, (3, 1), activation='relu'),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=5, batch_size=128, validation_split=0.2)
```

Conv2D: 필터 크기 (3×1), 특징 추출

MaxPooling2D: 중요 특징만 추림

Flatten: 1D 벡터로 변환

Dense: fully connected layer

Dense: sigmoid 활성화 → 이진 분류 확률 출력

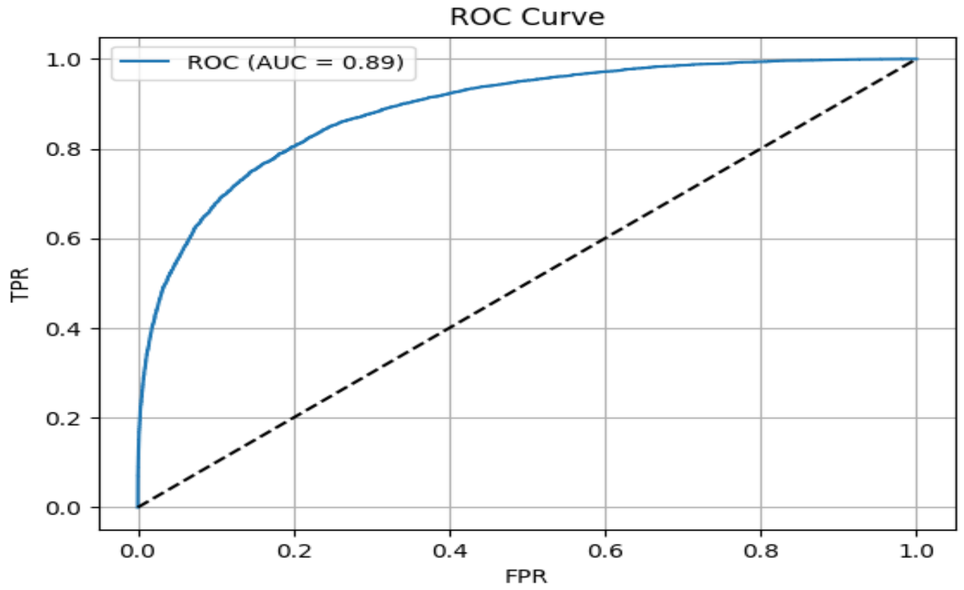
에폭 5회, 배치 128에 따라 학습 수행

예측 및 성능 평가

```
y_probs = model.predict(X_test).ravel()
fpr, tpr, _ = roc_curve(y_test, y_probs)
roc_auc = auc(fpr, tpr)

precision, recall, _ = precision_recall_curve(y_test, y_probs)
pr_auc = average_precision_score(y_test, y_probs)
```

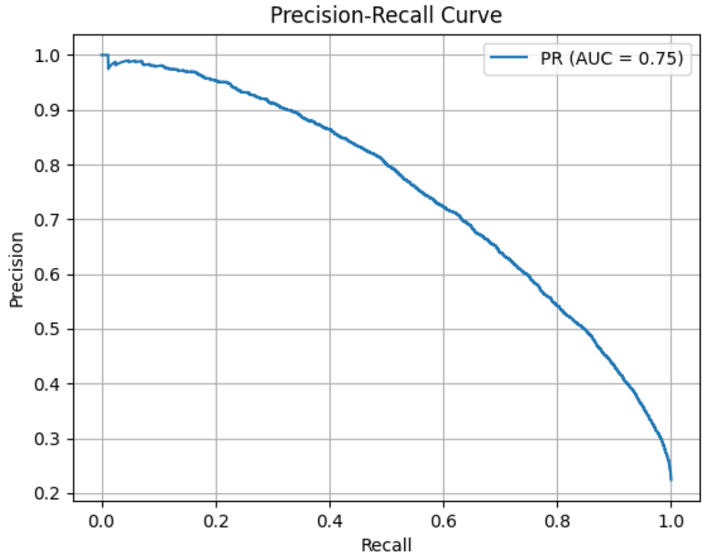
결과 및 성능



ROC 성능 분석 :

AUC = 0.89로 우수한 결과값을 가지고 있다.

비가 오늘 날에 대한 구분 능력이 우수하다.



PR 성능 분석 :

PR AUC = 0.75로 균형이 잘 잡힌 결과값을 가지고 있다.

데이터 불균형이 있을 경우 예측하는데 도움을 준다.

실제로 비가 오는날이 적기 때문에 데이터 불균형을 이루고 있다.