1. **전처리** 
   1. 결측치 제거 방식
      1. Age, Health score, Premium Amount 평균값 대체 : 소수의 결측치 이고 나이는 정규분포 따르니까
      2. Annual Income, Credit score 중앙값 대체 : 비대칭적인 분포를 갖기에 중앙값대체가 왜곡을 줄이고 이상치가 있을 수 있기에
      3. Marital Status 결측치 제거 : 범주형 변수이고 주요 피처가 아니라고 생각되어 삭제 고려 가능
   2. 범주형 인코딩
      1. Gender, Smoking : 카테고리가 두 개인 이진변수 이고, 하나의 열만 사용하기 위해 label encoding방식 사용
      2. Education, Exercise : 위계가 있는 순서형 변수로 운동을 자주 할 수록 건강할 가능성이 높고, 보험료를 낮게 책정할 수 있다고 의도적으로 세팅하기 위해 label encoding방식 사용
      3. Marital statis : 위계가 존재하지 않고 단순히 다른 상태이기 때문에 모델에 편향을 주지 않기 위해 one-hot encoding방식 사용
   3. 특징에 따른 분류

비슷한 성격 변수를 하나의 그룹으로 묶고 각 그룹별로 정규화 진행 후 평균을 통해 그룹스코어 생성

1. 개인정보 : 연령,성별,교육레벨
2. 건강관련 : 흡연,운동여부, 건강점수
3. 기타특성 : 보험지속기간, 신용점수
4. 각 그룹안에서 변수들을 동일한 기준으로 정규화시킴

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

# 1. Load dataset

df = pd.read\_csv("/Users/yuran/Documents/original\_dataset.csv")

# 2. Fill missing values (numeric)

df['Age'] = df['Age'].fillna(df['Age'].mean())

df['Health Score'] = df['Health Score'].fillna(df['Health Score'].mean())

df['Premium Amount'] = df['Premium Amount'].fillna(df['Premium Amount'].mean())

df['Annual Income'] = df['Annual Income'].fillna(df['Annual Income'].median())

df['Credit Score'] = df['Credit Score'].fillna(df['Credit Score'].median())

# 3. Drop missing in categorical columns (if they exist)

for col in ['Marital Status']:

if col in df.columns:

df = df.dropna(subset=[col])

# 4. Round Age

df['Age'] = df['Age'].round(1)

# 5. Lable encoding

education\_order = {'PhD': 0, "Master's": 1, "Bachelor's": 2, 'High School': 3}

exercise\_order = {'Daily': 0, 'Weekly': 1, 'Monthly': 2, 'Rarely': 3}

df['Gender'] = df['Gender'].map({'Male': 0, 'Female': 1})

df['Smoking Status'] = df['Smoking Status'].map({'No': 0, 'Yes': 1})

df['Education Level'] = df['Education Level'].map(education\_order)

df['Exercise Frequency'] = df['Exercise Frequency'].map(exercise\_order)

# 6. One-hot encoding - Marital Status

if 'Marital Status' in df.columns:

df = pd.get\_dummies(df, columns=['Marital Status'])

# 7. Define groups

personal\_features = ['Age', 'Gender', 'Education Level']

health\_features = ['Smoking Status', 'Exercise Frequency', 'Health Score']

other\_features = ['Insurance Duration', 'Credit Score']

# 8. Normalize within each group

scaler = MinMaxScaler()

df[personal\_features] = scaler.fit\_transform(df[personal\_features])

df[health\_features] = scaler.fit\_transform(df[health\_features])

df[other\_features] = scaler.fit\_transform(df[other\_features])

# 9. Group scores

df['Personal\_Score'] = df[personal\_features].mean(axis=1)

df['Health\_Score'] = df[health\_features].mean(axis=1)

df['Other\_Score'] = df[other\_features].mean(axis=1)

# 10. Save the cleaned and transformed dataset

df.to\_csv("Preprocessing\_Insurance\_Data.csv", index=False)

1. **클러스터링**

개인, 건강, 기타요인으로 분류하여 k-means clustering진행

유사한 그룹을 하나로 묶고 클러스터별 특성 파악하기 위해

import pandas as pd

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv("/Users/yuran/Documents/Preprocessing\_Insurance\_Data.csv")

# Personal score clustering

kmeans\_personal = KMeans(n\_clusters=3, random\_state=0)

df['Cluster\_Personal'] = kmeans\_personal.fit\_predict(df[['Personal\_Score']])

# visualization - Personal Score

plt.figure(figsize=(10, 5))

sns.boxplot(x='Cluster\_Personal', y='Personal\_Score', data=df)

plt.title("Personal Score by Cluster")

plt.grid(True)

plt.show()

# Health score clustering

health\_features = ['Smoking Status', 'Exercise Frequency', 'Health Score']

kmeans\_health = KMeans(n\_clusters=3, random\_state=42)

df['Cluster\_Health'] = kmeans\_health.fit\_predict(df[health\_features])

# visualization - Health Score

plt.figure(figsize=(10, 5))

sns.boxplot(x='Cluster\_Health', y='Health Score', data=df)

plt.title("Health Score by Cluster")

plt.grid(True)

plt.show()

# Other score clustering

other\_features = ['Insurance Duration', 'Credit Score']

kmeans\_other = KMeans(n\_clusters=3, random\_state=42)

df['Cluster\_Credit'] = kmeans\_other.fit\_predict(df[other\_features])

# visualization - Other Score

plt.figure(figsize=(10, 5))

sns.boxplot(x='Cluster\_Credit', y='Credit Score', data=df)

plt.title("Other Score by Cluster")

plt.grid(True)

plt.show()

1. Personal score

라인, 도표, 직사각형, 그래프이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Cluster0 : personal score 평균이 가장 높으며 고학력,고령,여성 중심

Cluster1 : personal score 평균이 가장 낮으며 분포가 넓고 극단적으로 낮은 값 또한 존재한다

Cluster2 : personal score이 중간수준으로 가장 고르게 퍼져있는 값

1. Health score

도표, 라인, 직사각형, 그래프이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Cluster0 : 중앙값이 약 0.3의 값을 갖고 하위 사분위수가 0에 가까워 일부 건강점수가 매우 낮음을 나타내고 흡연자 혹은 운동부족 포함 가능성이 존재

Cluster1 : 이상치가 다수 존재함 비흡연자여도 운동패턴이 다양하거나 건강 점수의 편차가 큼을 나타냄

Cluster2 : 중앙값이 세 그룹중 가장 높으며 전체적으로 건강점수가 균형잡혀있다 운동을 자주하거나 흡연자 비율이 적은 집단일 가능성이 보임

1. Other score

도표, 라인, 직사각형, 그래프이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Cluster0 : 분포가 넓고 0.3이상이 대부분인 값으로 신용이 비교적 좋은 고객그룹, 보험 가입 기간도 길 가능성 존재

Cluster1: 낮은 이상치 점수가 존재하여 신용점수는 중간수준,일부 위험고객이 포함된 그룹

Cluster2 : 중앙값이 매우 낮으며 신용점수가 매우 낮고 분포도가 좁아 신용 위험이 큰 고객 그룹일 가능성이 높고 보험료가 높게 측정될 것이라 예측

1. **클러스터링 기반 regression** – 두용님께서 작성해주신 자료 바탕
2. **결과 예측**

import pandas as pd

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import MinMaxScaler

# 1. Load the preprocessed dataset

df = pd.read\_csv("/Users/yuran/Documents/Preprocessing\_Insurance\_Data.csv")

# 2. Define features and target variable for model training

features = ['Age', 'Gender', 'Education Level', 'Smoking Status',

'Exercise Frequency', 'Health Score', 'Credit Score', 'Insurance Duration']

target = 'Premium Amount'

# 3. Normalize the feature values (using the same method as in preprocessing)

scaler = MinMaxScaler()

df[features] = scaler.fit\_transform(df[features])

# 4. Train the model

X = df[features]

y = df[target]

model = LinearRegression()

model.fit(X, y)

# 5. Get user input

print("Enter your information.")

age = float(input("Age (ex: 45): "))

gender = input("Gender (Male/Female): ")

edu = input("Education level (PhD, Master's, Bachelor's, High School): ")

smoke = input("Smoking status (Yes/No): ")

exercise = input("Exercise frequency (Daily, Weekly, Monthly, Rarely): ")

health\_score = float(input("Health score (0~100): "))

credit\_score = float(input("Credit score (0~1000): "))

duration = float(input("Insurance duration (year): "))

# 6. Encode categorical variables (consistent with original preprocessing)

gender = 0 if gender == "Male" else 1

smoke = 0 if smoke == "No" else 1

edu\_map = {'PhD': 0, "Master's": 1, "Bachelor's": 2, 'High School': 3}

exercise\_map = {'Daily': 0, 'Weekly': 1, 'Monthly': 2, 'Rarely': 3}

edu = edu\_map[edu]

exercise = exercise\_map[exercise]

# 7. Create a DataFrame for the input and apply the same scaling

user\_data = pd.DataFrame([[age, gender, edu, smoke, exercise, health\_score, credit\_score, duration]],

columns=features)

user\_data\_scaled = scaler.transform(user\_data)

user\_data\_scaled\_df = pd.DataFrame(user\_data\_scaled, columns=features)

# 8. Make prediction

predicted\_premium = model.predict(user\_data\_scaled\_df)

print(f"\nEstimated Premium Amount: {predicted\_premium[0]:,.2f}")

예시 1

텍스트, 폰트, 스크린샷이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

예시 2

텍스트, 폰트, 스크린샷이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

1. **End to end** // 1번의 전처리 과정과 비슷한 맥락이고 중간 내용도 위에서 설명하며 넘어와서 저희의 end to end 과정은 이러하다고 간결하게 적어도 괜찮을 것 같습니다.
   1. Objective settings : 보험 상품의 가격 결정에 있어, 고객의 다양한 특성을 고려한 맞춤형 보험료 책정은 보험사의 수익성 및 고객 만족도에 직결되는 중요한 요소로 본 프로젝트의 목적은 다음과 같다
      1. 보험 가입자의 개인,건강,재무 특성등을 기반으로 KMeans 클러스터링 사용하여 유사집단 도출
      2. 클러스터별로 맞춤형 회귀 모델을 학습하고 보험료 예측 정확도 비교
      3. 나아가 사용자 입력 기반 보험료 예측 기능을 구현하여 실용성 있는 도구로 발전 가능성 탐색
   2. Data curation
      1. 총 9개의 변수와 범주형/수치형 혼합 특성 존재
      2. 타겟 변수 : Premium Amount
   3. Data inspection
      1. 결측치 존재 : 수치형 변수 (Age, credit score 등), 범주형 변수 (Marital Status) 일부 누락
      2. 분포 결과 : Age, Health Score 대체로 정규 분포

Annual Income, Credit score 비대칭 분포->이상치 우려

* 1. Data Preparation
     1. 결측치 처리

평균 대체 : Age, Health Score, Premium Amount

중앙값 대체 : Annual Income, Credit score

제거 : Marital Status

* + 1. 인코딩 방식

Label Encoding : Gender, Smoking,E ducation level, Exercise frequency

One-hot Encoding : Marital Status

* + 1. 각 그룹 내 MinMax 정규화 + 평균값 산출
  1. Data Analysis
     1. Clustering

각 스코어에 대해 Kmeans Clustering 수행

* + 1. Regression

각 클러스터별로 선형 회귀 모델 학습

* 1. Evaluation
     1. 두용님이 보내주신 파일에서 지표가 MAE,RMSE,R^2 인 그래프 결과 적어주시면 좋을 것 같습니다.
  2. Development
     1. 사용자 특성입력을 기반으로 보험료 예측 기능 구현
     2. 전처리 및 정규화 -> Regression -> 예측 결과 출력

PPT 순서 // specification 참고해서 적었습니다. 꼭 이대로 안하고 기준에만 맞게 변경해주셔도 됩니다!

데이터 셋에 관한 설명 – 왜 이런 데이터셋을 골랐는지 (실생활에 사용 가능하다 등) proposal에 거의 적어 놨습니다 참고해서 밑부분의 데이터 분석에 대한 코드, 결과 사진도 포함 부탁드립니다

전처리 과정

클러스터링 과정

회귀 과정

결과 예측 과정

Teamwork data

short writeup on what you have learned

: 고객 특성 기반 스코어링과 클러스터링을 통해 데이터 그룹별 패턴을 탐색 할 수 있었고 이를 활용해 회귀 모델을 학습하여 보험료 예측을 시도하였다. 클러스터링을 통한 고객 세그먼트 파악은 의미 있었지만, 선형회귀의 한계로 인해 예측력은 제한적 이었다.

향후 과제 – 비선형 모델을 적용, 피처 엔지니어링을 강화 (상호작용 변수, 외부데이터 결합 등), 보험료에 영향 미치는 추가 변수 수집 등

Through score-based segmentation and clustering of customer characteristics, we were able to explore patterns within different data groups and attempted to predict insurance premiums using regression models. While clustering provided meaningful insights into customer segmentation, the predictive power was limited due to the constraints of linear regression.

**Future Work** – Apply nonlinear models, enhance feature engineering (e.g., interaction terms, integration of external data), and collect additional variables that may influence insurance premiums.

각 클러스터별로 개별 회귀 모델을 학습시키는 과정에서 대부분의 R² 성능이 예상보다 매우 낮게 나와 모델이 잘 작동하지 않는 이유를 분석해야 했습니다.

이를 해결하기 위해 MAE, RMSE 같은 다양한 평가지표를 추가하고, 실제값과 예측값을 시각화하여 클러스터별로 모델의 작동 방식을 비교했습니다. 그 결과, 클러스터가 나뉘었다고 해도 그것이 항상 예측 성능 향상으로 이어지지 않는다는 점을 배울 수 있었습니다.

While training separate regression models for each cluster, I noticed that the R² scores were much lower than I expected, so I had to figure out why the models weren’t working well.

To fix this, I added other evaluation metrics like MAE and RMSE, and also visualized the actual and predicted values to see how each model behaved in different clusters. Through this, I learned that just splitting the data into clusters doesn’t always mean the prediction will be better.

발표, 질문 시간이 짧아서 코드와 사진등으로 채우면 잘 넘어갈 수 있을 것 같습니다..