옷 사이즈 예측하기

컴퓨터공학과 17학번 김형근

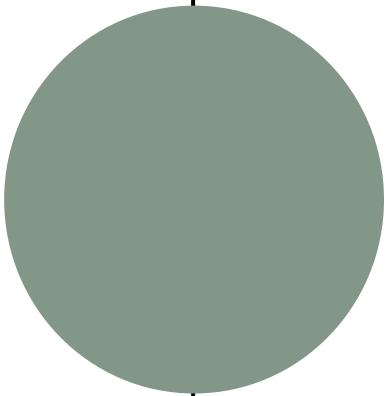
옷 사이즈 예측하기

목 차

코드 설명

모델 정확도 비교

느낀 점



옷 사이즈 예측하기

개요

사람의 체중(weight), 나이(age), 키(height)값을 이용하여

옷의 사이즈(size)를 예측하는 코드를 분석합니다.

또한 여러 인공지능 모델의 정확도를 비교합니다.

옷 사이즈 예측하기

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data = pd.read_csv(f)
print(data)
```

```
        weight   age  height size
0           62  28.0  172.72   XL
1           59  36.0  167.64    L
2           61  34.0  165.10    M
3           65  27.0  175.26    L
4           62  45.0  172.72    M
...        ...   ...     ...  ...
119729      63  42.0  175.26    M
119730      45  29.0  154.94    S
119731      61  31.0  172.72    M
119732      74  31.0  167.64   XL
119733      70  30.0  167.64   XL

[119734 rows x 4 columns]
```

import로 모듈을 입력합니다.

csv 파일을 읽을 수 있도록 합니다.

[119735 x 4]
0 ~ 119734. 약 12만개의 데이터가 있습니다.

weight, age, height는 수치로. Size는 (XXS, S, M, L, XL, XXL, XXXL) 7종으로 분류됩니다.

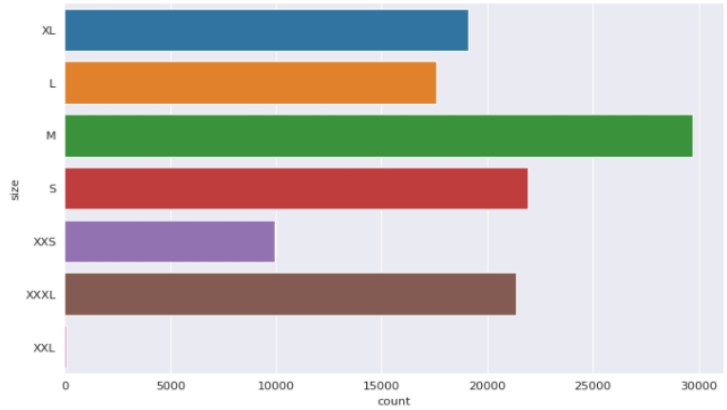
옷 사이즈 예측하기

사이즈 분포 확인하기

```
data[["size"]].value_counts()
```

```
size
M       29712
S       21924
XXXL    21359
XL      19119
L       17587
XXS      9964
XXL        69
dtype: int64
```

```
plt.figure(figsize=(10, 6), dpi=80)
sns.countplot(y=data["size"])
plt.show()
```



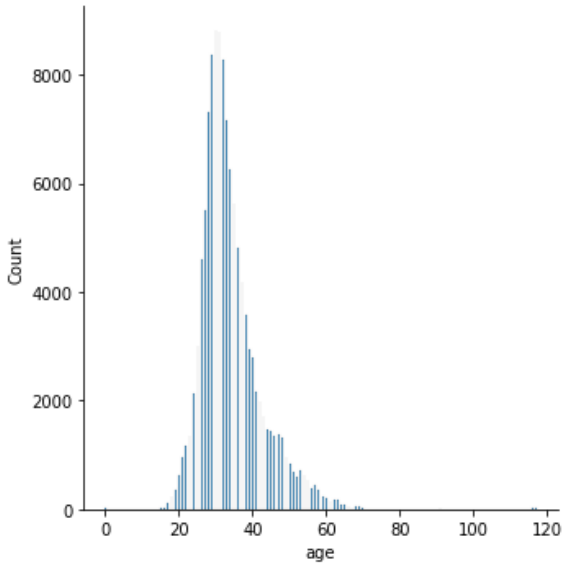
옷 사이즈 예측하기

나이 분포 확인하기

```
# Age distribution
sns.displot(df_raw["age"])
```

나이의 분포를
displot으로 나타내기.

20 ~ 40(세) 사이 높게
집계되었습니다.



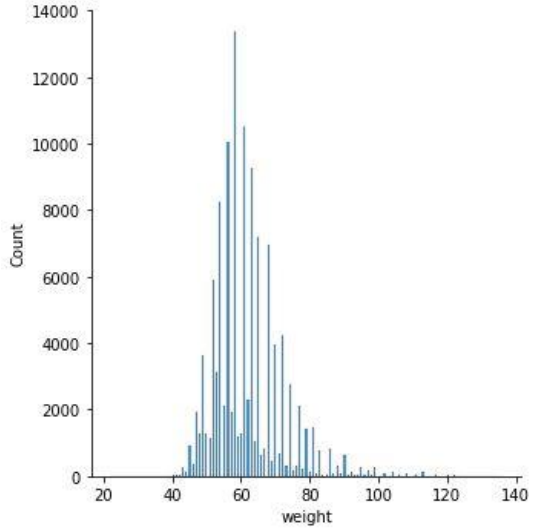
옷 사이즈 예측하기

몸 무게 분포 확인하기

```
# Weight distribution
sns.displot(df_raw["weight"])
```

몸 무게의 분포를
displot으로 나타내기.

40 ~ 80 (KG) 사이에 높게
집계되었습니다.



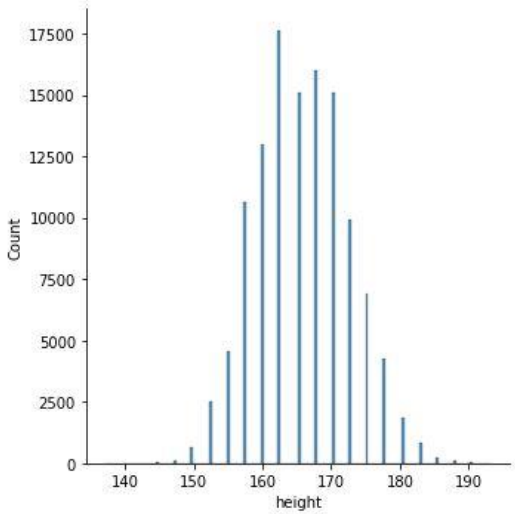
옷 사이즈 예측하기

키 분포 확인하기

```
# height distribution
sns.displot(df_raw["height"])
```

키의 분포를
displot으로 나타내기.

155 ~ 175(CM) 사이에 높게
집계되었습니다.



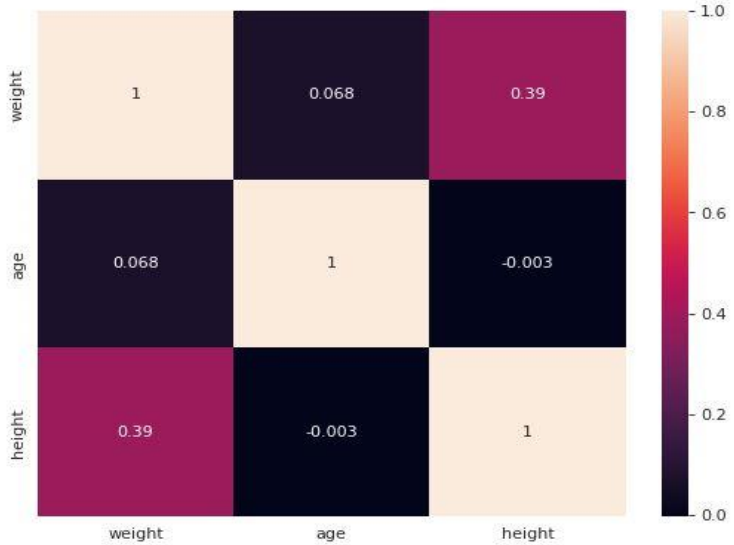
옷 사이즈 예측하기

히트맵 연관성 확인하기

```
plt.figure(figsize=(8, 6), dpi=80)
sns.heatmap(data.corr(), annot=True)
plt.show()
```

히트맵을 사용하여 두 개의 카테고리 값에 대한 값 변화를 알아보겠습니다.

weight와 height 사이에 높은 수치가 나왔으며,
age와 height 사이에 가장 낮은 수치가 나왔습니다.



옷 사이즈 예측하기

결측 값 검색하기

```
print(data.isnull().sum())
```

```
weight      0
age       257
height    330
size        0
dtype: int64
```

X, Y 값에 대입하기 위해 결측이 있는지 검색합니다.
데이터 값 중 결함이 있는 부분을 탐색하고 나타냅니다.

옷 사이즈 예측하기

X 에 데이터 입력하기

```
from sklearn.impute import SimpleImputer

imp = SimpleImputer(missing_values=np.nan, strategy='mean')
X = data[['weight','age','height']]
X = imp.fit_transform(X)
print(X)
```

```
[[ 62.    28.   172.72]
 [ 59.    36.   167.64]
 [ 61.    34.   165.1 ]
 ...
 [ 61.    31.   172.72]
 [ 74.    31.   167.64]
 [ 70.    30.   167.64]]
```

평균을 통해 결측 값 채우기 - (결측 값은 숫자)
Mean => 평균값

옷 사이즈 예측하기

Y 에 데이터 입력하기

```
from sklearn.preprocessing import OneHotEncoder, LabelEncoder

y = np.array(data[['size']]).ravel()
le = LabelEncoder()
le_y = le.fit_transform(y)
print(le_y)
```

```
[3 0 1 ... 1 3 3]
```

범주형 클래스를 숫자로 변환 - 레이블 인코더

옷 사이즈 예측하기

분류하기

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.33, random_state=0)

print(X_train)
print(y_train)

print(X_test)
print(y_test)
```

[x, y]각각
훈련용(train), 테스트용(test)으로 스플릿 후 표현.
각종 모델에 대입해보도록 하겠습니다.

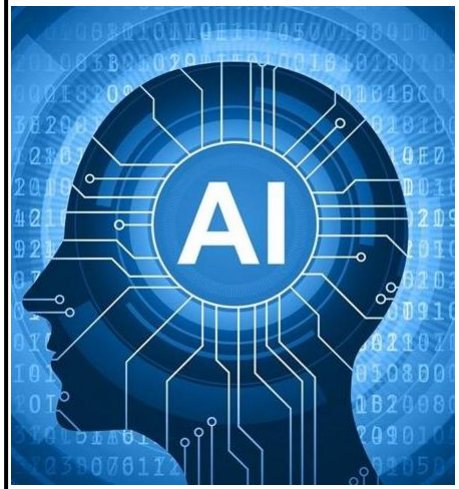
```
[[ 63.    25.   177.8 ]
 [ 63.    33.   175.26]
 [ 56.    49.   160.02]
 ...
 [ 68.    40.   175.26]
 [ 47.    29.   157.48]
 [ 68.    30.   162.56]]
['L' 'M' 'M' ... 'XXXL' 'XXS' 'XL']
[[ 63.         22.         162.56     ]
 [ 61.         45.         162.56     ]
 [ 50.         68.         165.1      ]
 ...
 [ 72.         34.0273107 170.18     ]
 [ 72.         41.         170.18     ]
 [ 54.         23.         177.8      ]]
['L' 'L' 'M' ... 'L' 'XL' 'S']
```

옷 사이즈 예측하기

모델의 정확도 비교

여러 인공지능 모델의 정확도를 비교합니다.

1. Logistic Regression (로지스틱 회귀) 모델
2. Naïve Bayes (나이브 베이즈) 모델
3. Stoachastic Gradient Descent (확률적 경사 하강법) 모델
4. Decision Tree (결정 트리) 모델
5. Random Forest 모델
6. Support Vector Machine 모델
7. 총 평가



옷 사이즈 예측하기

1. Logisitic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

lr = LogisticRegression(solver="liblinear").fit(X_train, y_train)
y_pred = lr.predict(X_test)
accuracy_lr = accuracy_score(y_test, y_pred)
print("Accuracy for Logistic Regression: %.2f" % accuracy_lr)
```

```
Accuracy for Logistic Regression: 0.47
```

종속변수가 범주형 이면서 0 or 1 인 경우 사용하는 분석입니다.

Logisitic Regression 모델의 정확도를 계산합니다.

옷 사이즈 예측하기

2. Naïve Bayes

```
from sklearn.naive_bayes import GaussianNB

nb = GaussianNB()
nb.fit(X_train, y_train)
y_pred = nb.predict(X_test)
accuracy_nb = accuracy_score(y_test, y_pred)
print("Accuracy for Naive Bayes: %.2f" % accuracy_nb)
```

```
Accuracy for Naive Bayes: 0.48
```

Bayes 법칙에 기반한 분류기 혹은 학습 방법입니다.

Naïve Bayes 모델의 정확도를 계산합니다.

옷 사이즈 예측하기

3. Stoachastic Gradient Descent

```
from sklearn.linear_model import SGDClassifier
sgd = SGDClassifier(loss='modified_huber', shuffle=True, random_state=0)
sgd.fit(X_train, y_train)
y_pred = sgd.predict(X_test)
accuracy_sgd = accuracy_score(y_test, y_pred)
print("Accuracy for Stochastic Gradient Descent: %.2f" % accuracy_sgd)
```

```
Accuracy for Stochastic Gradient Descent: 0.42
```

확률적 경사 하강법으로 데이터 세트에서 무작위로 균일하게 선택한 하나의 예를 의존하여, 예측 경사를 계산합니다.

Stoachastic Gradient Descent 모델의 정확도를 계산합니다.

옷 사이즈 예측하기

4. Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(max_depth=5, random_state=0, max_features=None, min_samples_leaf=
5)
dtree.fit(X_train, y_train)
y_pred = dtree.predict(X_test)
accuracy_dt = accuracy_score(y_test, y_pred)
print("Accuracy for Decision Tree: %.2f" % accuracy_dt)
```

```
Accuracy for Decision Tree: 0.51
```

일련의 분류 규칙을 통해 데이터를 분류, 회귀하는 지도 학습 모델 중 하나이며,
결과 모델이 Tree 구조를 가지고 있기 때문에 Decision Tree라는 이름을 가집니다.

Decision Tree모델의 정확도를 계산합니다.

옷 사이즈 예측하기

5. Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rfm = RandomForestClassifier(n_estimators=50, oob_score=True, n_jobs=3, random_state=0, max_feat
ures=None, min_samples_leaf=15)
rfm.fit(X_train, y_train)
y_pred = rfm.predict(X_test)
accuracy_rfm = accuracy_score(y_test, y_pred)
print("Accuracy for Random Forest: %.2f" % accuracy_rfm)
```

```
Accuracy for Random Forest: 0.52
```

랜덤 포레스트는 훈련을 통해 구성해놓은 다수의
나무들로부터 분류 결과를 취합해서 결론을 얻는 방식입니다.

Random Forest모델의 정확도를 계산합니다.

옷 사이즈 예측하기

6. Support Vector Machine

```
from sklearn.svm import SVC
svm = SVC(kernel="linear", C=1, random_state=0)
svm.fit(X_train,y_train)
y_pred=svm.predict(X_test)
accuracy_svm = accuracy_score(y_test, y_pred)
print("Accuracy for Support Vector Machine: %.2f" % accuracy_svm)
```

```
Accuracy for Support Vector Machine: 0.49
```

두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적인 이진 선형 분류 모델을 만듭니다.

Support Vector Machine 모델의 정확도를 계산합니다.

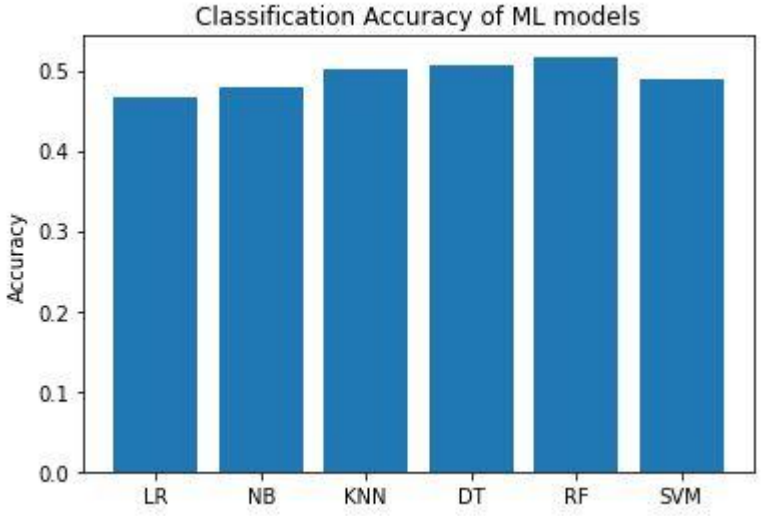
옷 사이즈 예측하기

7.1 평가 (코드)

```
import matplotlib.pyplot as plt

labels = ['LR', 'NB', 'KNN', 'DT', 'RF', 'SVM']
accuracies = [accuracy_lr, accuracy_nb, accuracy_knn, accuracy_dt, accuracy_rfm, accuracy_svm]

x = [0,1,2,3,4,5]
width=0.35

fig, ax = plt.subplots()
ax.bar(x=labels,height=accuracies)

ax.set_ylabel('Accuracy')
ax.set_title('Classification Accuracy of ML models')
ax.set_xticks(x)
ax.set_xticklabels(labels)

plt.show()
```

옷 사이즈 예측하기

7.2 평가 (결과)



LR = Logistic Regression
NB = Naïve Bayes
KNN = K-Nearest Neighbor
DT = Decision Tree
RF = Random Forest
SVM = Support Vector Machine

Random Forest의 모델이 가장 높은 정확성을 보입니다.

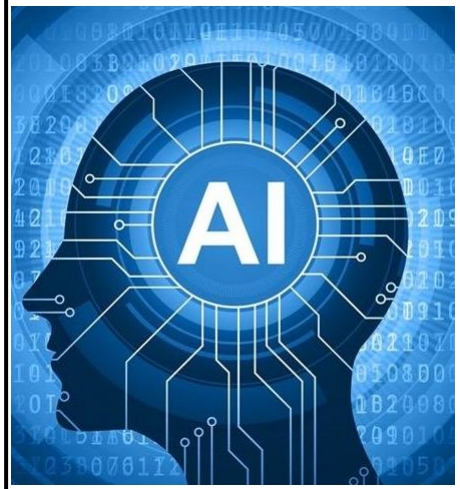
옷 사이즈 예측하기

느낀점

이번 발표를 준비하면서 이 과제를 조사하며 코드의 전체적인 큰 내용은 이해가 가능했지만, 세부적으로 코드를 조사하면서 각종 인공지능 모델에 대해서 이해하기 어려운 점이 있었습니다.

70퍼센트 정도 코드를 이해한 것 같습니다.

수업 내용과 겹치는 내용이 있어서 복습하는 느낌으로 공부를 진행했던 것 같습니다.



옷 사이즈 예측하기

감사합니다.