뜨거운 예측, 쿨한 드라이빙

(Hot Predictions, Cool Drives)

학번: 2018016
이름: 류경용
Github address
https://github.com/2018016-Kyungyong-Ryu/Hot_Predictions-Cool_Drives

1. 안전 관련 머신러닝 모델 개발의 목적

a. 학습 모델 활용 대상

전기자동차의 동작중 여러 가지 변수들중 모터의 성능 저하를 일으키는데 연관된 변수가 어떤것인지를 파악하여 다음과 같은 상황에서 활용할 수 있다.

1). 예측 및 경고시스템

머신러닝 모델을 사용하여 전기자동차의 모터 성능의 저하와 여러 변수들 사이의 관계를 학습한 후, 이를 기반으로 성능이 저하되는 상황에서 운전자에게 경고를 줄 수 있다. 이를통해 운전자에게 사고의 위험에 대한 사전 정보를 제공하고, 조치를 취하도록 할 수 있다.

2). 효율적인 전력 사용

모델을 활용하여 특정 주행 조건에서의 최고의 효율이 나는 조건을 예측함으로써, 전력사용을 최적화할 수 있다. 이를 통해 전기자동차의 주행 효율을 향상시키고 배터리 수명을 연장할 수 있다.

b. 데이터에서 사용할 독립변수와 이를통해 예측할 종속 변수

데이터에서 사용된 변수는 총 13개로, u_q, 냉각제 온도, 고정자 권선의 온도, u_d, 고정자 톱니 온도, 모터 속도, i_d, i_q, 영구자석 온도, 고정자 요크 온도, 주변 온도, 토크, profile_id가 있다. 이중 u-q, u-d, I-d, I-q에 따라 모터 속도와 토크가 바뀌게 되고, 모터 속도와 토크가 바뀜에 따라 고정자 톱니 온도, 모터 속도, 영구자석 온도, 고정자 요크 온도가 바뀌게 된다.

모터의 성능과는 영구자석의 온도와 관련이 있다. 만약 영구자석의 온도가 올라가게 된다면 성능의 백분율을 잃으므로 영구자석의 온도를 낮게 유지하는 것이 모터의 성능을 유지하는데 중요하다. 따라서 종속변수를 영구자석의 [온도(pm)]으로 설정하고 해당 변수와 연관성이 깊은 [u-d, 고정자의 톱니온도, 모터의 속도, i_d, 모터 요크의 온도, 주변온도, 토크] 를 독립변수로 설정하였다.

c. 개발의 의의

해당 머신러닝 모델을 사용하여 지속적인 전기자동차를 개발하여 현재 전기자동차의 문제인 배터리 수명, 화재의 위험성을 현저히 낮출 수 있을 것 이다.

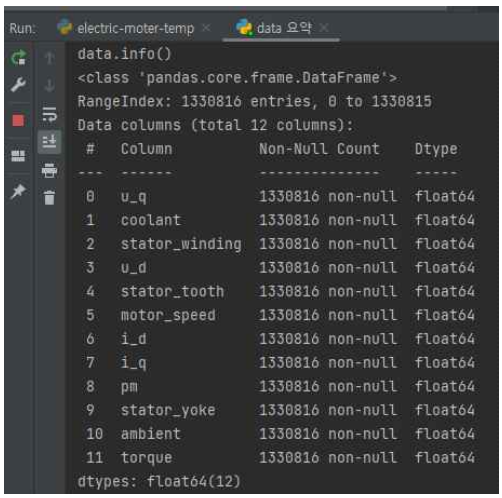
2. 안전 관련 머신러닝 모델의 네이밍의 의미

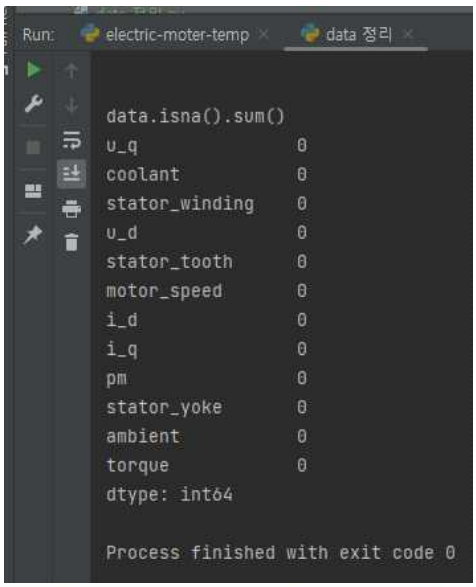
a. 모델에서 가장 중요하게 생각하는 것은 모터의 영구자석의 열 예측이다. 이러한 열 예측을 통해 궁극적으로 운전자의 편의와 안전을 도모하는 것을 목표로 삼았으며, 이를 제목을 통해 중요한내용으로 강조하였다.

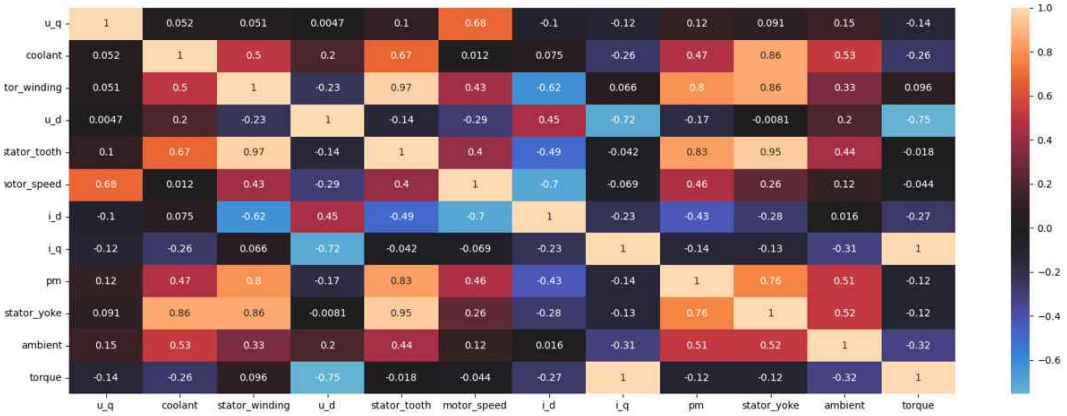
3. 개발 계획

a. 데이터에 대한 요약 정리 및 시각화

기존의 데이터가 너무 방대하여 노트북의 파이참으로 데이터를 요약할 수가 없어 데이터의 크기를 줄여 요약을 하였다.





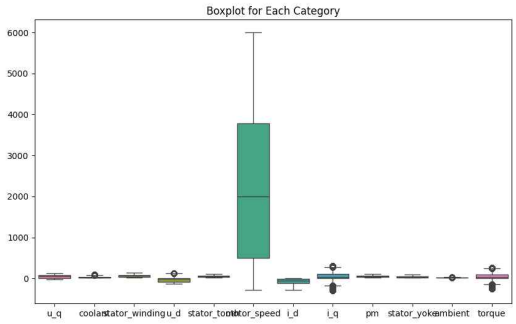


b. 데이터 전처리 계획

데이터를 확인해본 결과 우선 데이터의 크기가 매우 컸다. 따라서 데이터의 크기를 줄여 용량을 줄인 다. 그런 뒤 데이터 요약을 통해 확인한 각 변수들의 이상치를 제거하여 더욱 정확한 데이터를 만든다.

c. 사용할 머신러닝 모델

사용할 머신러닝 모델은 선형 회귀(Linear Regression)을 사용할 것이다.



이러한 모델을 선정한 이유로는 종속변수인 영구자석의 온도(pm, X)가 다른 독립변수들(고정자의 온도, 주변온도 등, Y)의 영향을 받아 선형적인 변화를 나타낼것을 heatmap을 통해 확인하였기 때문이다.

$$y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_n \cdot x_n$$

선형 회귀 모델은 일반적으로 위의 형태를 갖는다. y는 예측하려는 변수이고, x는 입력 변수이며, b는 모델에 기여하는 계수이다.

선형 회귀 모델의 특징으론 첫째, 모델이 입력변수에 대해 선형으로 변하는 것으로 가정되며, 둘째, 계수들이 직관적으로 해석 가능하다는 장점이 있다. 또한 상대적으로 간단한 모델이기에 학습과 예측이 빠르며, 대용량 데이터셋에서도 잘 작동할 수 있다.

이와같은 특징을 바탕으로 만들고자 하는 모델을 학습시키기에 적합한 모델로 판단하여 선형 회귀 모델을 선택하였다, 모델의 예측이 성공적이라면 그래프는 우상향 대각선으로 나타나게 된다.

d. 머신러닝 모델 예측 결과

연관관계를 나타내는 heatmap에서 볼 수 있듯 독립변수와 종속변수가 양의 상관관계로 이루어져있다. 즉 영구자석의 온도는 권선의 온도, 고정자 톱니의 온도, 고정자 요크의 온도, 모터의 속도, 토크에 비례해 오르는 선형그래프로 나타날 것 이다.

e. 사용할 성능 지표

사용할 성능지표는 평균 제곱 오차(Mean Squared Error, 이하 MSE)를 사용할 것 이다. 이는 머신 러닝 모델의 예측값과 실제값 간의 차이를 측정하는 지표중 하나이다. 주로 회귀(Regression)모델을 통해 예측한 모델에 사용하는 지표이다.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

MSE는 수학적으로 다음과 같이 정의하며 여기서 n은 데이터 포인트 개수를, y_i는 실제 값을, $\hat{y}_i$는 모델의 예측값을 나타낸다. MSE는 제곱을 함으로 음수 값이나 큰 오차에 더욱 민감하게 반응한다. 이러한 MSE는 값이 작을수록 모델의 예측이 실제 값과 가깝다고 해석할 수 있지만, 주관적인 기준

에 따라 작다는 기준은 달라질 수 있다.

f. 성능 검증 방법 계획 등

MSE를 통해 평균값을 구하여 값이 20보다 작다면 예측이 잘 되었다고 판단하도록 한다. 또한 MSE의 결과를 산점도를 통해 그려 시각화를 통해 성능을 검증한다.

4. 개발 과정

a. 학습모델 개발 과정

우선 기본이 되는 데이터프레임의 크기가 너무 크므로 데이터를 잘라 크기를 줄인다. 과제 제출시에는 너무 큰 데이터가 Github에 업로드 되지 않으므로 해당 과정은 과제에서 제외하도록 한다.

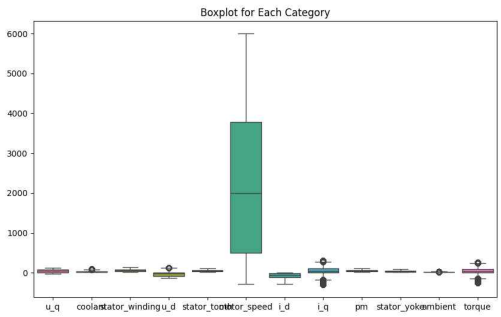
해당 과정을통해 데이터를 10개로 나눈 뒤, 각 그룹에서 30퍼씩 지움으로 데이터의 편향을 방지하고, 크기를 줄였다.

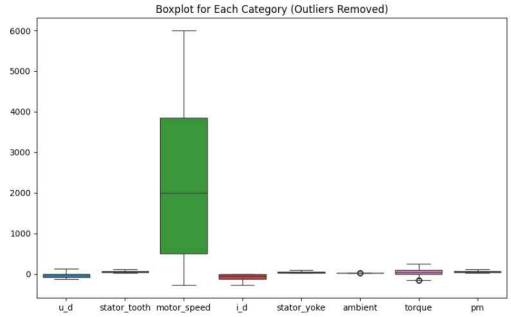
데이터를 확인했을 때, 결측치가 존재하지 않았으므로 데이터의 빈공간을 채우는 과정은 생략하였다. 또한 사용하기로 한 독립변수와 종속변수를 따로 추출해 다시 저장하기로 한다.

```
index = [3, 4, 5, 6, 9, 10, 11,  8]
df = data.iloc[:, index]
print(df.head(5))
print(df.columns)
```

```
        u_d  stator_tooth  motor_speed  ...    ambient    torque         pm
0 -0.350055     18.293219     0.002866  ...  19.850691  0.187101  24.554214
1 -0.305803     18.294807     0.000257  ...  19.850672  0.245417  24.538078
2 -0.372503     18.294094     0.002355  ...  19.850657  0.176615  24.544693
3 -0.316199     18.292542     0.006105  ...  19.850647  0.238303  24.554018
4 -0.332272     18.291428     0.003133  ...  19.850639  0.208197  24.565397

[5 rows x 8 columns]
Index(['u_d', 'stator_tooth', 'motor_speed', 'i_d', 'stator_yoke', 'ambient',
       'torque', 'pm'],
      dtype='object')
```

인덱스를 기준으로 사용할 변수들을 추출한 데이터 프레임으로 다시한번 박스플롯을 그려 이상치의 제거가 잘 되었는지 확인한다.





데이터 전처리 전 데이터 전처리 후

전처리가 끝난 데이터프레임을 확인하였다면 train과 test를 나눈다음 선형회귀모델을 사용해 머신러닝 코드를 작성한다.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=41)
model = LinearRegression()
model.fit(X_train, Y_train)

Y_pred = model.predict(X_test)
print(Y_pred)
```

인공지능 모델의 작성이 끝난뒤 정확도를 판단하기 위해 mse를 구하는 코드를 작성한다

```
Y_pred = model.predict(X_test)
print(Y_pred)

mse = mean_squared_error(Y_test, Y_pred)
print("Mean Squared Error (MSE):", mse)
```

구한 mse를 기준으로 산점도 및 회귀선을 시각화하여 보여주는 코드를 작성한다.

```
# 산점도 및 회귀선 시각화.
plt.figure(figsize=(10, 6))
plt.title('Actual vs Predicted Values')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
mse = sns.regplot(x=Y_test, y=Y_pred, scatter_kws={'s': 30, 'alpha':0.5}, line_kws={'color': 'red'})
mse.get_figure().savefig("./result/mse.png")
plt.show()
```

b. 각 함수는 어떻게 동작하는 지 구체적으로 설명

```
corr = data.corr("pearson")
```

사용된 해당 코드는 데이터프레임의 각 열 간의 피어슨 상관계수를 계산하여 반환하는 코드로 heatmap을 사용해 사용자에게 각 변수들간 상관관계가 얼마나 높은지를 보여줄 수 있다.

```
for column in df.columns:
        Q1 = df[column].quantile(0.25)
        Q3 = df[column].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        df = df[(df[column] >= lower_bound) & (df[column] <=
        upper_bound)]
```

사용된 해당 코드는 주어진 데이터프레임 df에서 각 열의 이상치를 제거하는 코드이다. for문을 사용해 데이터프레임의 각 열에 대해 반복하도록 하고, 현재 열의 1사분위수와 3사분위수를 계산을 한 뒤 이상치의 하한 경계와 상한 경계를 계산해 경계 내에 있는 행들만을 선택하여 이상치를 제거하도록 하는 코드이다.

```
mse = mean_squared_error(Y_test, Y_pred)
print("Mean Squared Error (MSE):", mse)
```

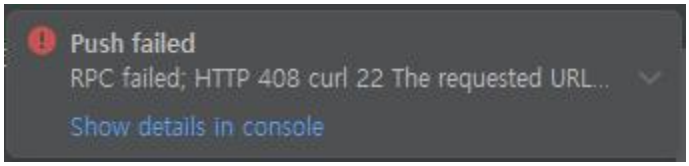
사용된 해당 코드는 머신러닝으로 학습된 모델의 정확성을 측정하기 위해 mse를 계산하는 함수로, 계산한 mse를 print()를 통해 보여준다.

c. 에러 발생지점

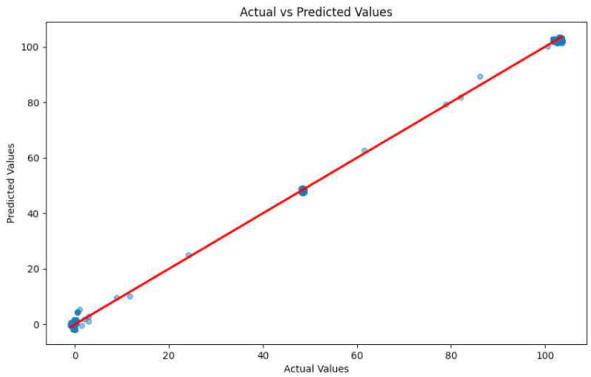
```
Y_pred = model.predict(X_test)
print(Y_pred)

accuracy = accuracy_score(Y_test, Y_pred)

print("Accuracy : ", accuracy)
```

```
ValueError: continuous-multioutput is not supported
```

코드를 실행하고 정확도를 검증하던 중 다음과같은 오류가 발생하였다. 이는 정확도를 검증하는 방법중 하나인 accuracy_score을 사용하여서였는데, 해당 검증방법은 분류를 하는 모델을 검증할 때 사용하는 방법이었다. 즉 주어진 변수의 선형적인 관계를 예측하는 본 모델에서는 어울리지 않는 검증방식이므로 좀 더 잘 어울리는 평균 제곱 오차(MSE)를 사용하기로 하였다.



깃헙에 푸쉬중 푸쉬에 실패했다는 오류가 출력되었다. 이는 Github에 업로드하는 파일의 용량이 매우 커서 발생한 오류이므로 사용하는 데이터프레임의 크기를 1,000,000개로 줄여 프로그램을 실행하도록 하였다.



데이터를 축소한 결과 데이터의 편향이 감지되었다. 따라서 데이터를 축소하는 과정을 중간부터 마지막까지 자르는 방식이 아닌 데이터를 총 10개의 그룹으로 나눈 뒤, 각 그룹에서 30퍼센트씩 축소하기로 하였다.

d. 학습 모델의 성능 평가

회귀 모델의 정확성을 나타내는 징표인 평균 제곱 오차(MSE)를 사용하여 정확성을 측정해본 결과 Mean Squared Error (MSE): 12.260210029202272 정도로 비교적 작은 숫자를 기록하 였다. 이러한 작은 숫자를 기록할 수 있었던 이유로는 두가지를 들 수 있을 것 같다. 첫째 구하고자 하는 종속변수와 연관성이 큰 독립변수를 지정하였다. 둘째 표본의 수가 많아 학습에 더 유리하였다. 하지만 예측에 실패한 결과또한 존재하는데 이 이유로는 데이터의 이상치가 존재하였던 것이 이유인 것 같다.

5. 개발후기

개발을 진행하기위해 사용한 데이터의 크기가 컸던게 큰 어려움중 하나였다. 데이터가 크니 육안으로 파악하지 못하고 파이썬을 통해서만 확인을 하였고, 이로인해 데이터를 전처리하기 위해 파악하기에 어려움을 겪었다. 하지만 장점또한 있었는데 바로 heatmap을 통해 데이터간의 유사성을 파악하기 편했고, 이러한 장점은 변수를 설정하는데 편리함을 주었다. 데이터를 축소할 때 느꼈던 어려운점은 단순히 데이터를 반으로 축소하다보니 머신러닝을 하기위한 데이터가 편향되는 것을 느꼈고, 이를 해결하기 위한 방법을 알아내기 또한 힘들었다. 이러한 어려움들을 통해 데이터를 가지고 머신러닝을 할 때 결과에 가장 큰 영향을 주는 요소들을 파악할 수 있게 되었고, 머

신러닝의 이해를 더욱 잘할 수 있었게 되었다.

e. 결과 시각화