Machine Learning 2024 Summer HW 2

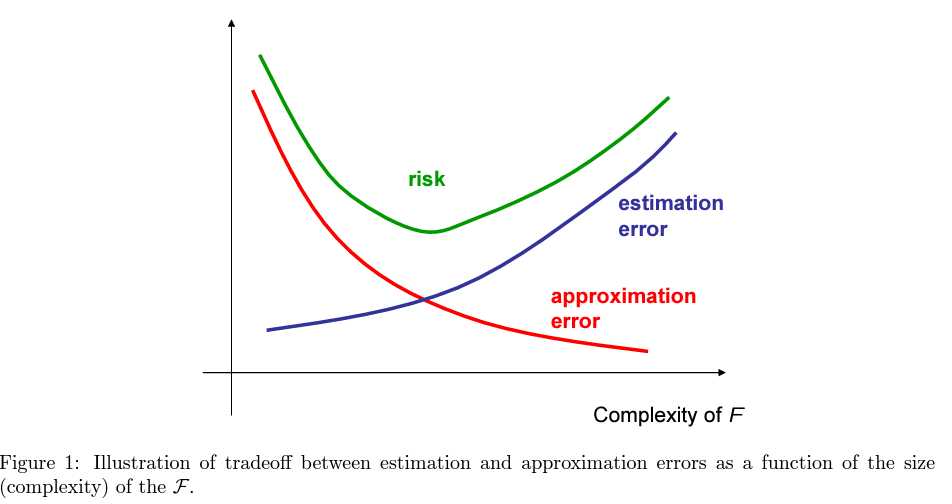
서울대학교 뇌인지과학과 박준오

OOO

1. 사전 조사 – 박준오

Model을 통해 데이터를 predict하는 것은 반드시 bias-variance tradeoff가 존재합니다. 일반적으로 데이터의 오류를 나타내는 loss function인 MSE를 전개하면 irreducible bias+bias^2 + variance의 형태로 나타납니다. 여기서 irreducible bias는 말 그대로 모델을 통해 바꿀 수 없는 bias이고, bias와 variance는 모델의 특성에 따라 변할 수 있습니다. Bias란 정답이 되는 데이터에서 얼마나 편향되게 벗어나는지에 대한 정도라고 할 수 있고, variance란 얼마나 무작위하게 퍼지는지를 나타내는 정도라고 할 수 있습니다.

출처: 위키피디아, ‘Bias-Variance tradeoff’, <https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff>



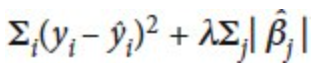
출처: University of Wisconsin-Madison, <https://nowak.ece.wisc.edu/SLT07/lecture3.pdf>

Bias-Variance trade-off figure 와 같이, model complexity가 증가할수록 approximation error, 즉 variance는 감소하게되지만, estimation error인 bias는 증가하게 됩니다. 이는 overfitting이 되는 것과 동치입니다. 반대로, model compexity가 감소할수록 variance는 증가하고 bias는 감소하게 됩니다. 이것은 underfitting이 되는 것과 동치입니다.

따라서 이 사이 적절한 모델을 찾는 것이 성능(approximation의 영향)과 일반화(estimation의 영향) 모두에 좋은 기대를 할 수 있습니다. 이것을 가능하게 해주는 것이 바로 regularization입니다.

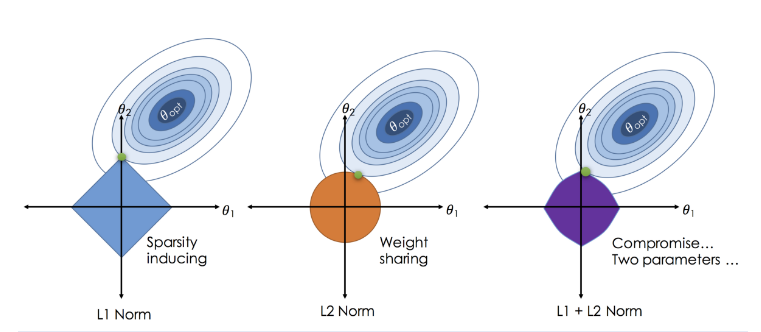
Regularization의 원리는, 각 parameter의 magnitude에 규제를 걸어주고자 loss function에 penalty term을 추가하는 것입니다. L2 norm에 대한 penalty term을 넣으면 MSE+L2 penalty 는

와 같이 정의할 수 있습니다. 람다가 커질수록 페널티가 커지게 됩니다. 마찬가지로, L1 norm에 대한 penalty term을 넣으면 MSE+L1 penalty는

와 같이 정의할 수 있습니다. 이 두가지 텀을 모두 넣는 것을 elasticnet이라고 하며, 두가지 람다로 페널티가 조절됩니다.



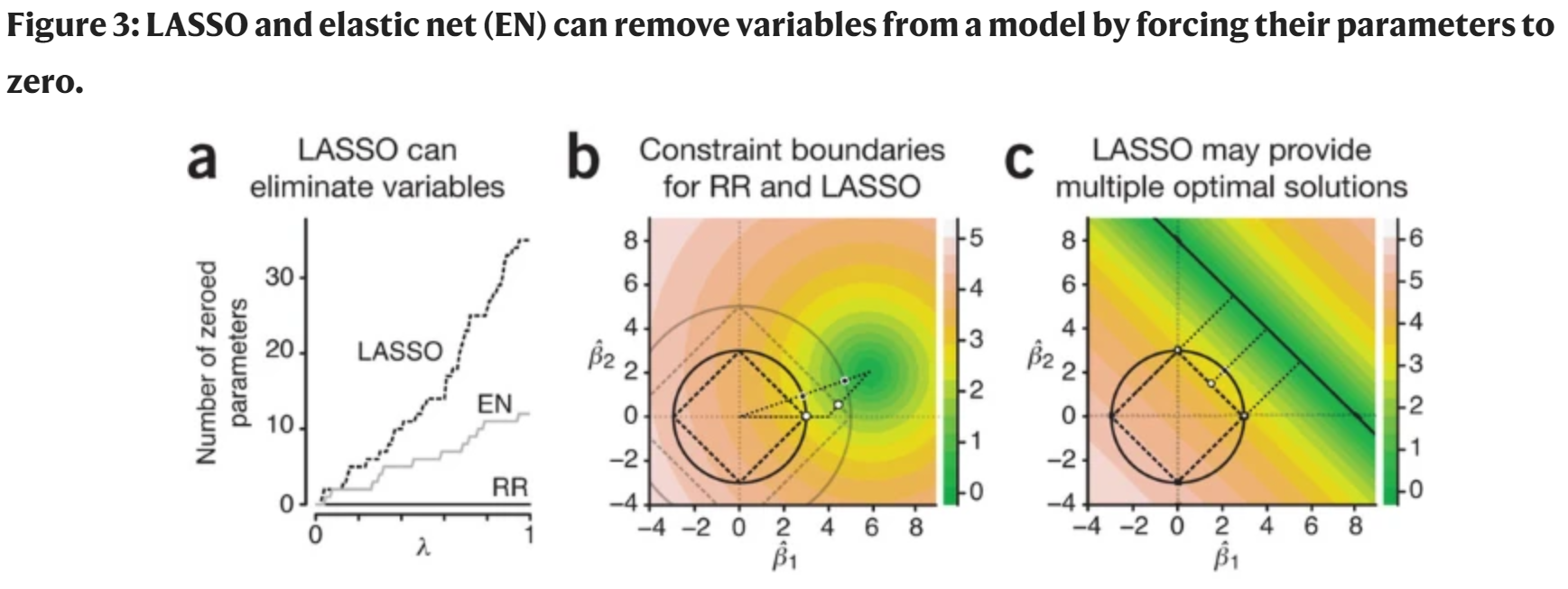
시각화하면 다음과 같습니다.



출처: Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net.

예를 들어 Y~b1X1+b2X2+b3X3의 회귀식에서 X1과 X2에 상관관계가 X1=2X2와 같이 존재할 때, b2 = -2b1이라는 무한한 계수 합이 존재할 수 있습니다. 그러나 이 회귀식은 결국 Y~0X1+0X2+b3X3이라는 간단한 모델로 나머지 독립변수를 제거할 수 있습니다. 이러한 복잡성을 줄이기 위해 regularization이 필요하며, Ridge regularization(L2) 을 통해 규제하게되면 각 b1과 b2의 크기가 감소하여 X1과 X2의 영향을 줄이게 되고, Lasso regularization(L1)을 통해 규제하게되면 각 X1와 X2를 완전히 소거하여 영향을 없에게 됩니다.

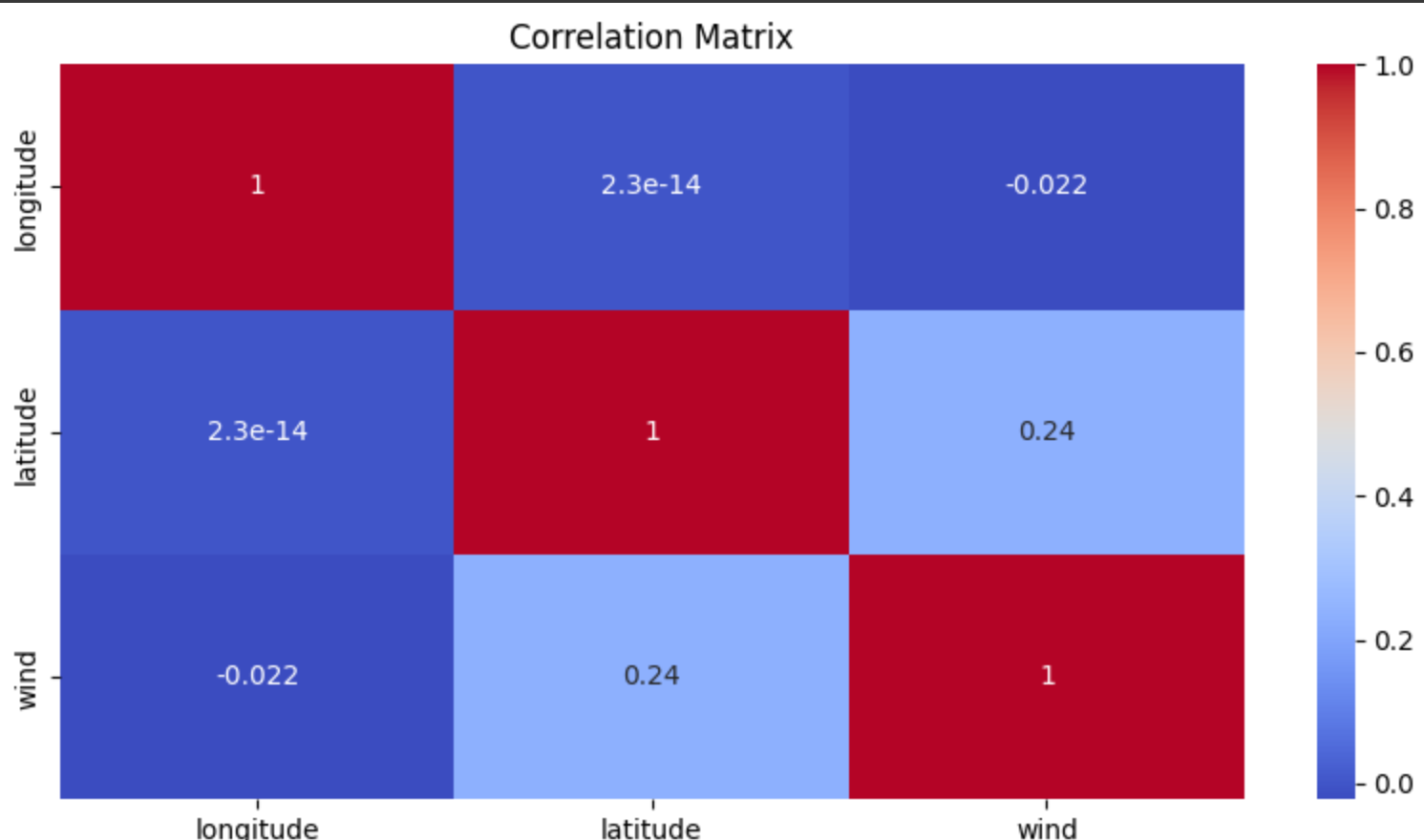
Ridge의 경우 parameter를 완전히 소거하지는 못하지만 Lasso의 경우 완전히 소거할 수 있고, 단점으로는 L1 norm의 한계로 unique solution을 갖지 못하는 경우도 있습니다. 둘을 합친 Elasticnet의 경우 람다가 2개 존재하기 때문에 최적해를 찾고자 여러가지 combination set을 구성해야하며, Lasso보다 독립변수 제거능력이 떨어질수도 있습니다.

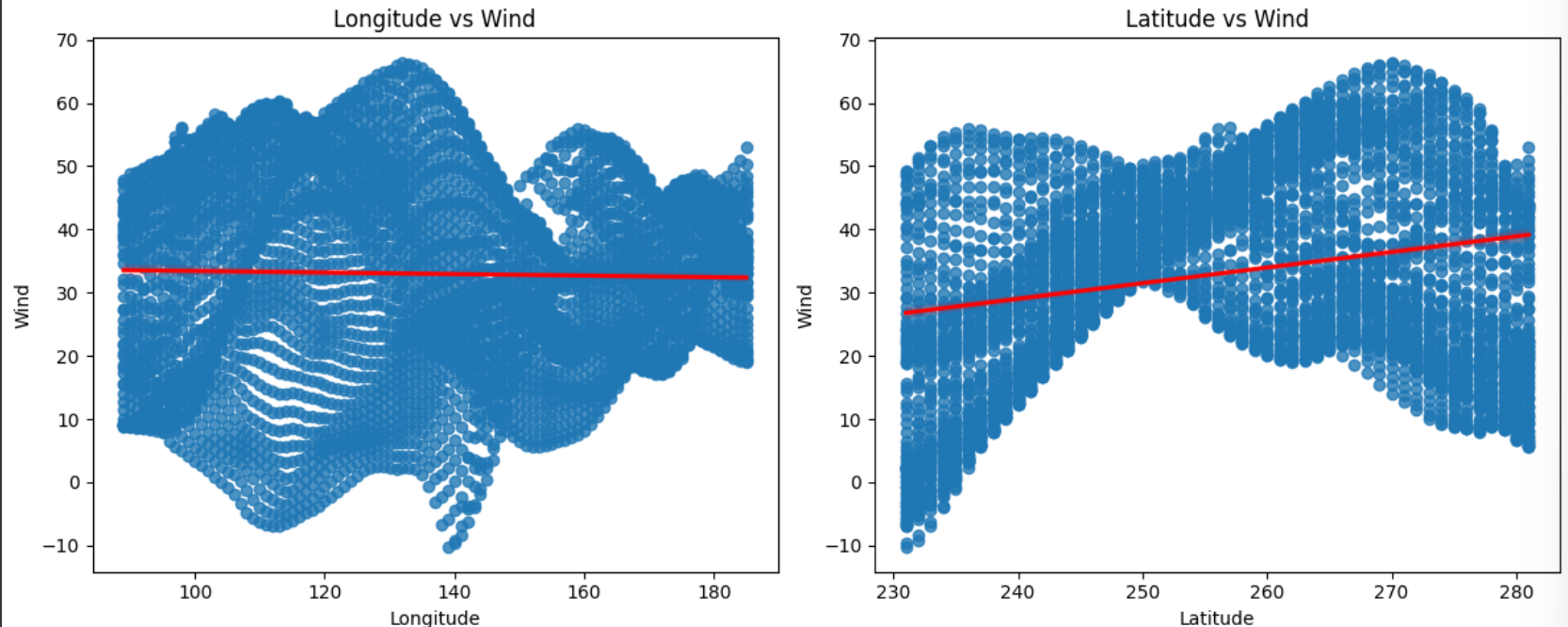


출처: Lever, J., Krzywinski, M. & Altman, N. Regularization. *Nat Methods* **13**, 803–804 (2016)

2. 데이터 전처리 및 기본적 검증 – 박준오, 박유나 코드

위도와 경도는 국제적으로 약속된 독립변수임이 틀림 없지만, 회귀의 안정성을 위협할 수 있는 interaction을 확인하고자 먼저 위도와 경도의 correlation을 확인했습니다. 회귀결과 두 변수간 correlation이 거의 없음을 확인했습니다. 회귀의 안정성을 위협할 수 있는 outlier를 제거하고자 zscoring을 통해 데이터를 normalize한 다음 3 sigma rule에 따라 멀리 떨어진 데이터를 제거하였습니다.



이후 X1과 X2중 하나만으로 Y를 predict할 수 있는지 확인해보았습니다. 

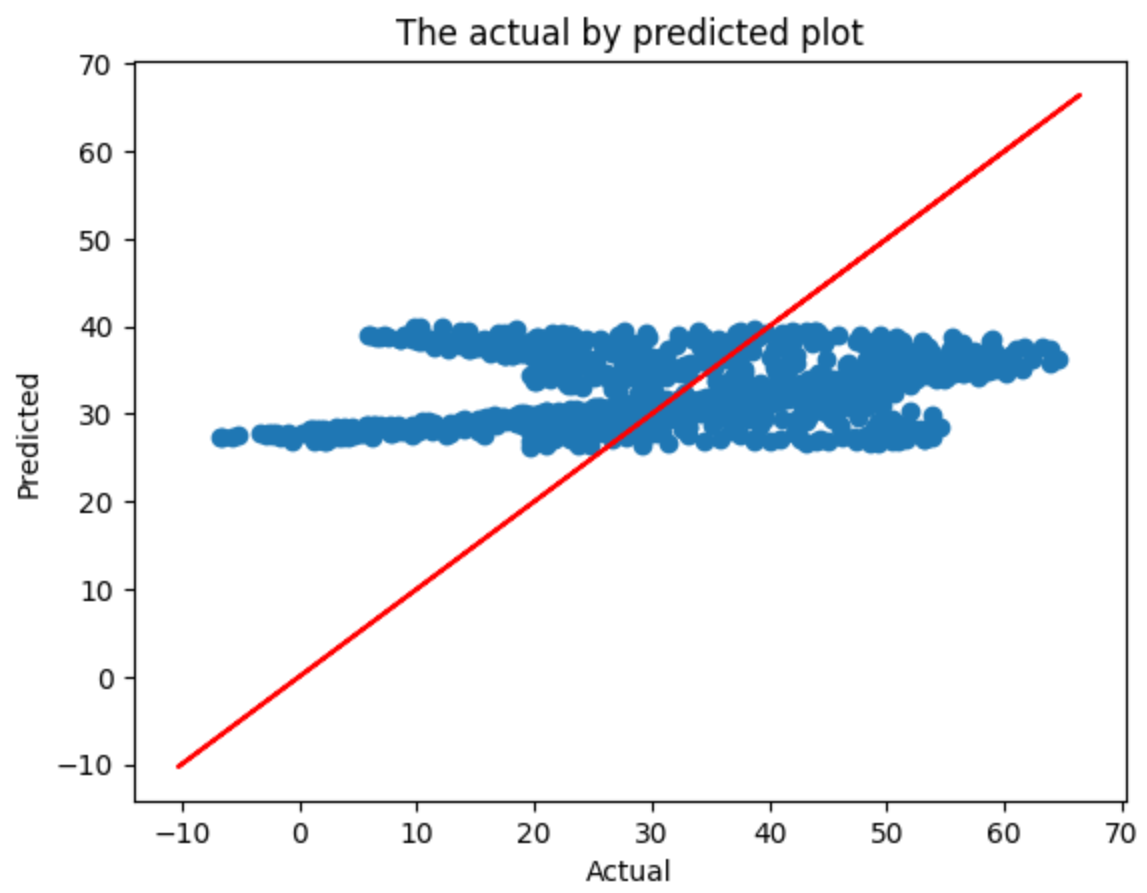
빨간색 회귀선에 파란색 데이터들이 전혀 fit되지 않았음을 확인할 수 있습니다.

3. 단순 회귀 모델 – 박유나 코드

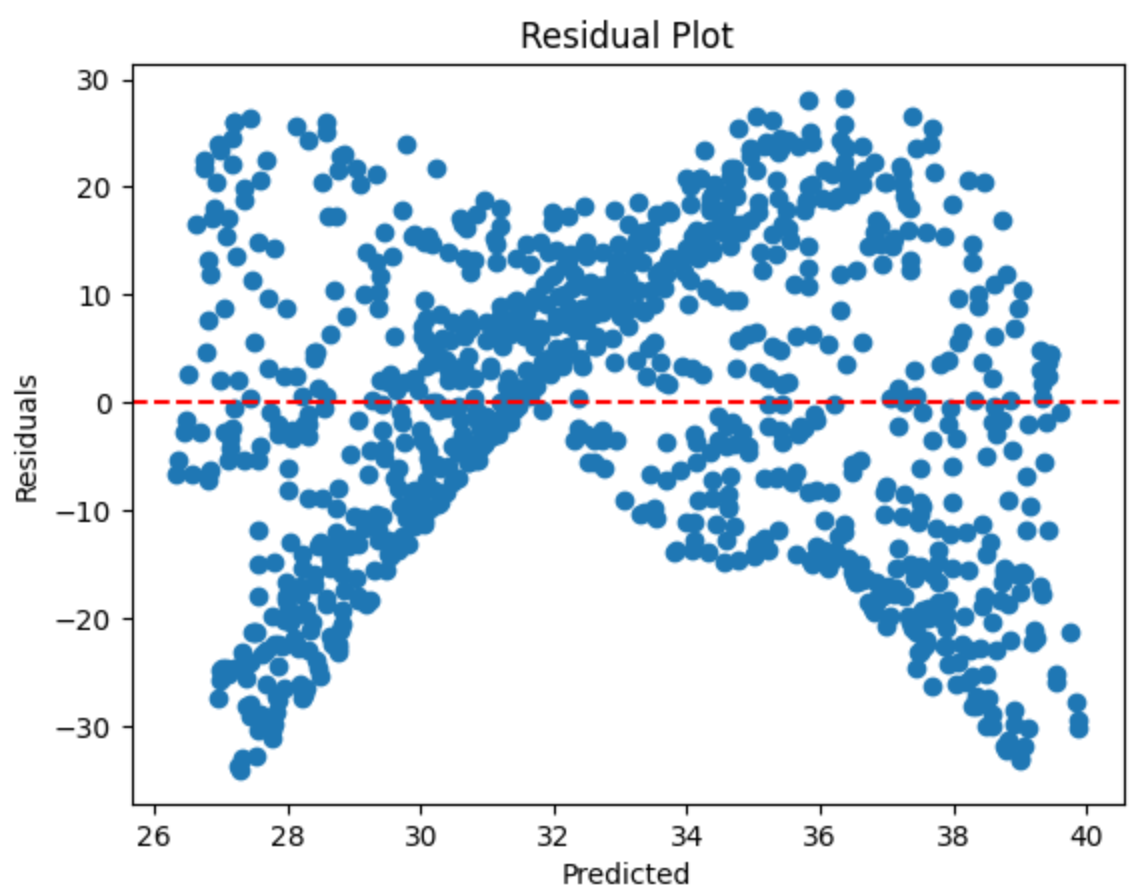
과제의 회귀 문제를 해결하기 위해 우선 MLP가 아닌 선형 회귀를 이용해 보았습니다. 스케일링과 이상치 제거가 끝난 데이터셋을 sklearn 패키지에서 제공하는 train\_test\_split 함수를 이용해 훈련 데이터와 테스트 데이터로 8:2 비율로 나누었습니다. 독립변수들로부터 종속변수를 설명하기 위한 모델로 선형 회귀 모델을 선택하였고, 이를 위해 sklearn 패키지에서 제공하는 LinearRegression() 함수를 사용했습니다. 훈련 데이터를 이용한 모형 적합 결과는 다음과 같습니다.

Linear Regression Equation: Eastward wind = 33.09428353 - 0.37778567 \* Longitude + 3.62083257 \* Latitude

단순 선형 회귀 모델을 평가해 보았습니다. 먼저, 결정 계수(R²) 값은 0.06으로, 1에 가까울수록 모델이 잘 적합되었다고 볼 수 있는데, 이 값은 상당히 낮습니다. 따라서 모델이 종속변수와 설명변수의 관계를 잘 설명하지 못한다고 할 수 있습니다.



실제값과 예측값의 관계를 나타내는 plot에서 붉은 선은 이상적인 예측 선입니다. 대부분의 점들이 이상적인 예측 선을 기준으로 고르게 배열되지 않고, 그 선과 멀리 분포해 있습니다. 이는 실제값과 예측값의 차이가 크다는 것을 의미하므로 모델이 잘 적합되지 않았다는 것을 보여줍니다.



다음으로, residual plot에서 모델이 잘 적합되고 예측을 잘 한다면, 잔차들이 0을 기준으로 랜덤하게 분포되어 있어야 합니다. 그러나 해당 그림에서는 잔차들이 0과 매우 떨어져서 분포되어 있으며, 나비 모양으로 커졌다 줄어드는 일정한 패턴이 관측됩니다. 따라서 모델이 잘 적합되지 않았다고 할 수 있습니다.

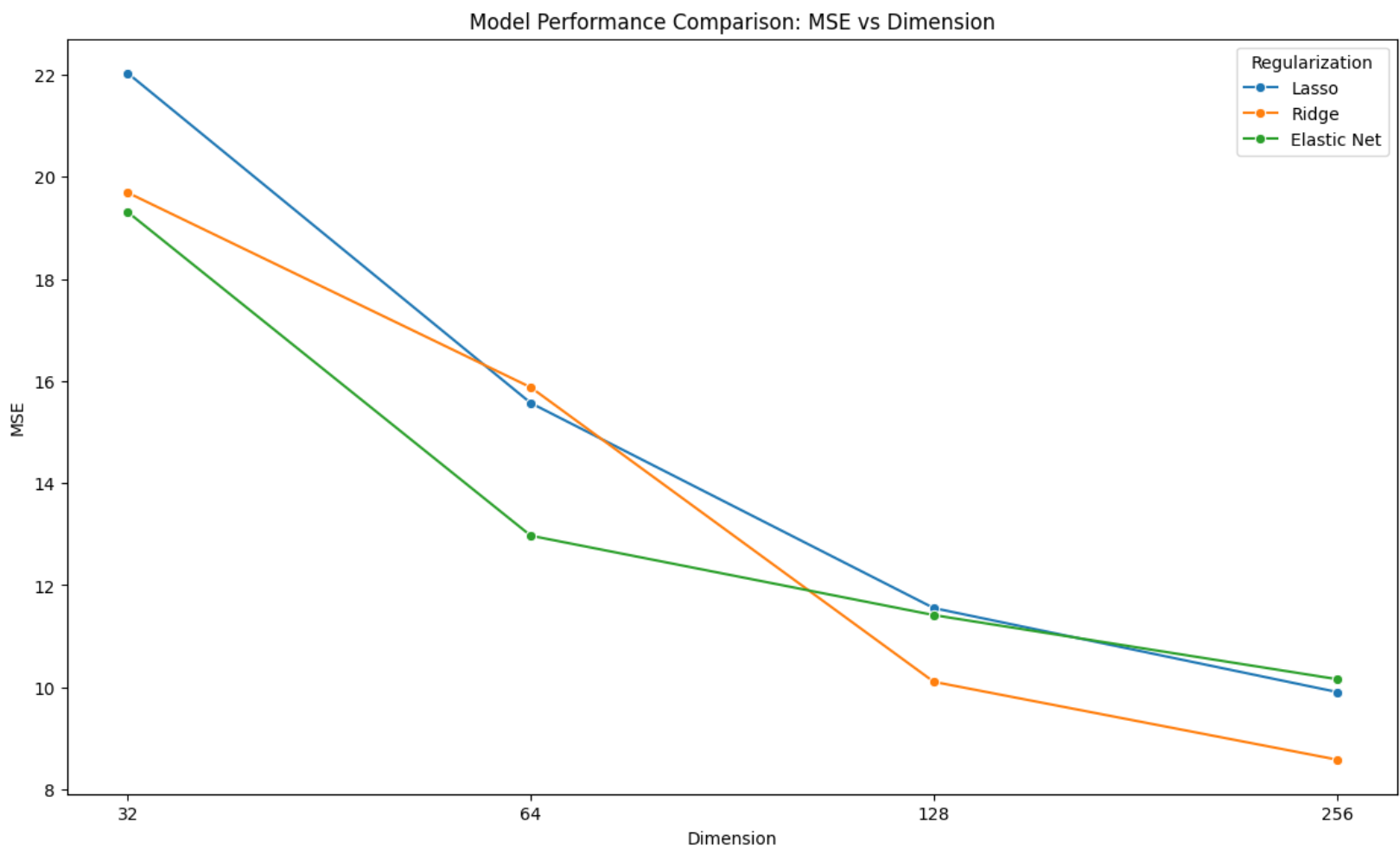
마지막으로, MRE를 계산하기 위해 모델 적합에 사용하지 않은 테스트 데이터셋에 대해 RMSE 값을 구했습니다. 해당 값은 14.9로, 값이 낮을수록 모델이 잘 예측한다고 볼 수 있는 것을 고려할 때, 이 값은 상당히 높은 수치로 모델의 설명력이 낮음을 의미합니다.

3. MLP Based Regression Model & Evaluation – 박준오 코드, 박유나 조사

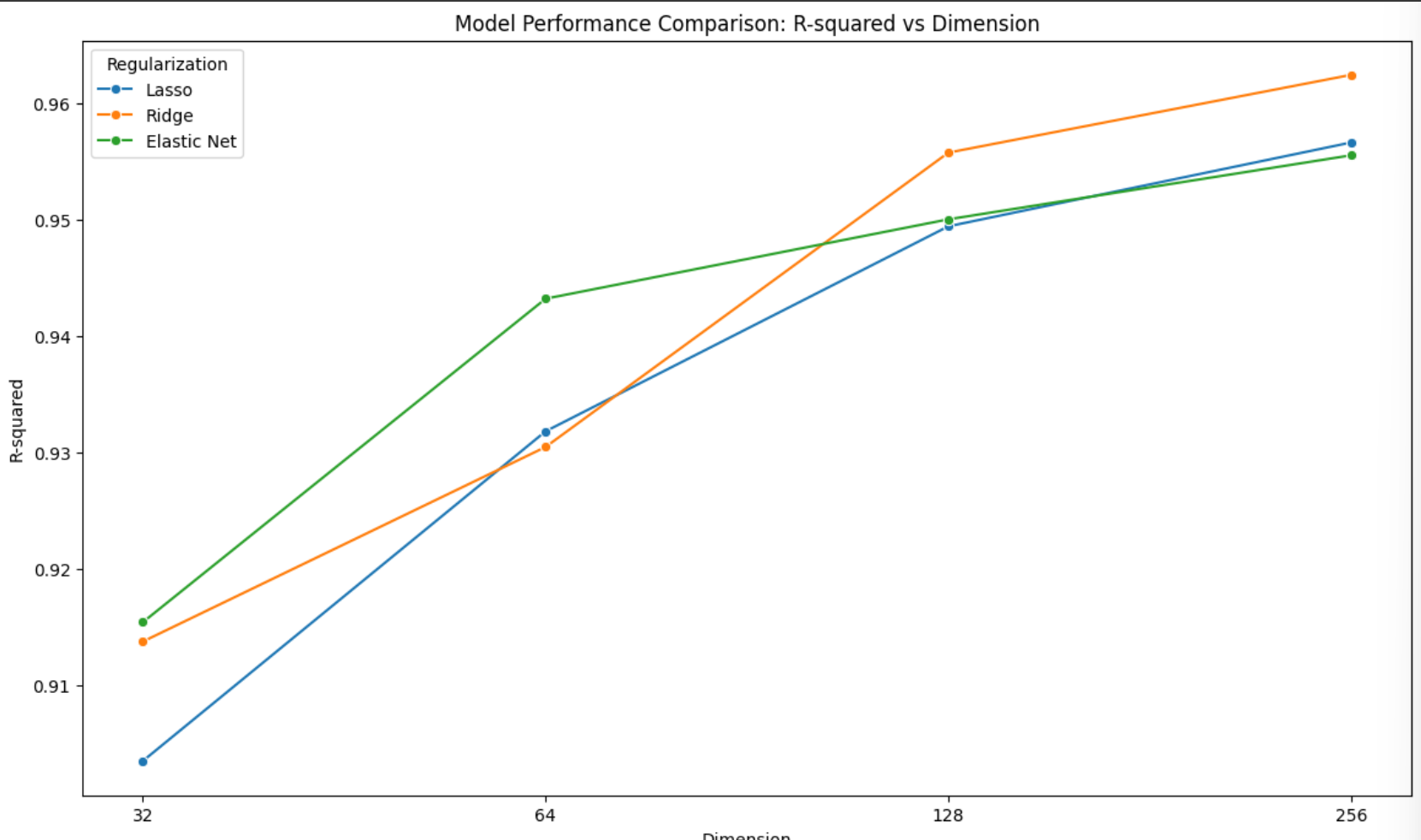
“Jeff Heaton. 2008. Introduction to Neural Networks for Java, 2nd Edition (2nd. ed.). Heaton Research, Inc.”에 따르면, 두 개의 히든 레이어가 있는 신경망은 어떤 형태의 함수든 표현할 수 있기 때문에 두 개가 넘는 레이어를 사용할 필요가 없다고 합니다. 따라서 2개의 히든 레이어를 갖도록 모델을 구성했습니다. (조사 박유나)

앞서 조사한 3가지 regularization(Ridge, Lasso, Elasticnet)을 hidden layer의 dimension에 따라 구성해보았습니다. Loss function에 penalty term을 더해주는 방식으로 코드를 짜보았습니다.

노드의 개수가 적으면 언더피팅이 발생하고, 노드가 너무 많으면 오버피팅이 발생하며 학습에 걸리는 시간이 늘어날 수 있습니다. 따라서 노드의 개수를 변경해가며 모델의 MSE와 R-squared을 평가했습니다. 다음은 [32, 64, 128, 256] 노드의 개수를 x축에, MSE와 R-squared를 y축에 3가지 regularization(lamda = 0.01)에 따라 나타낸 것입니다.

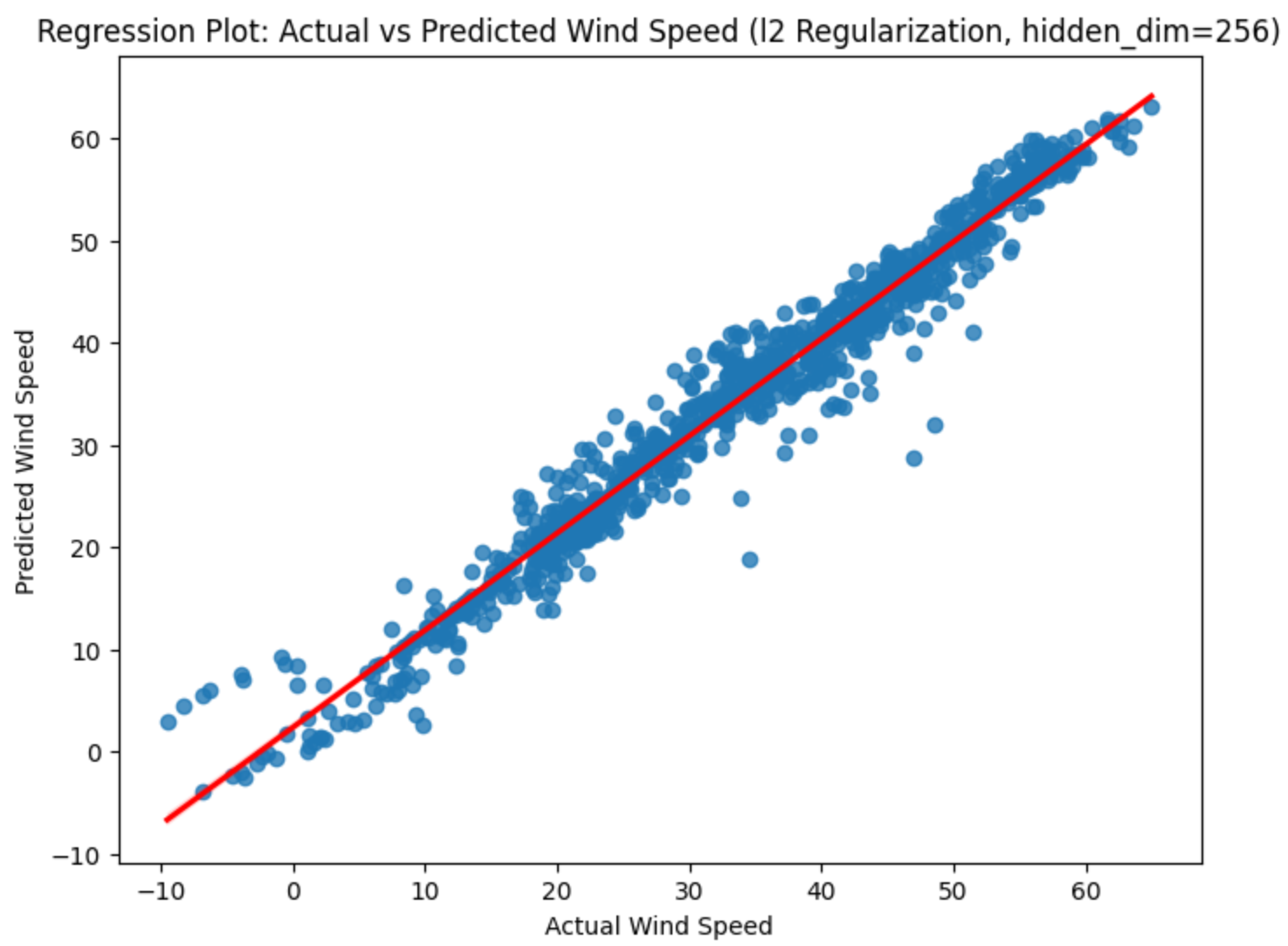


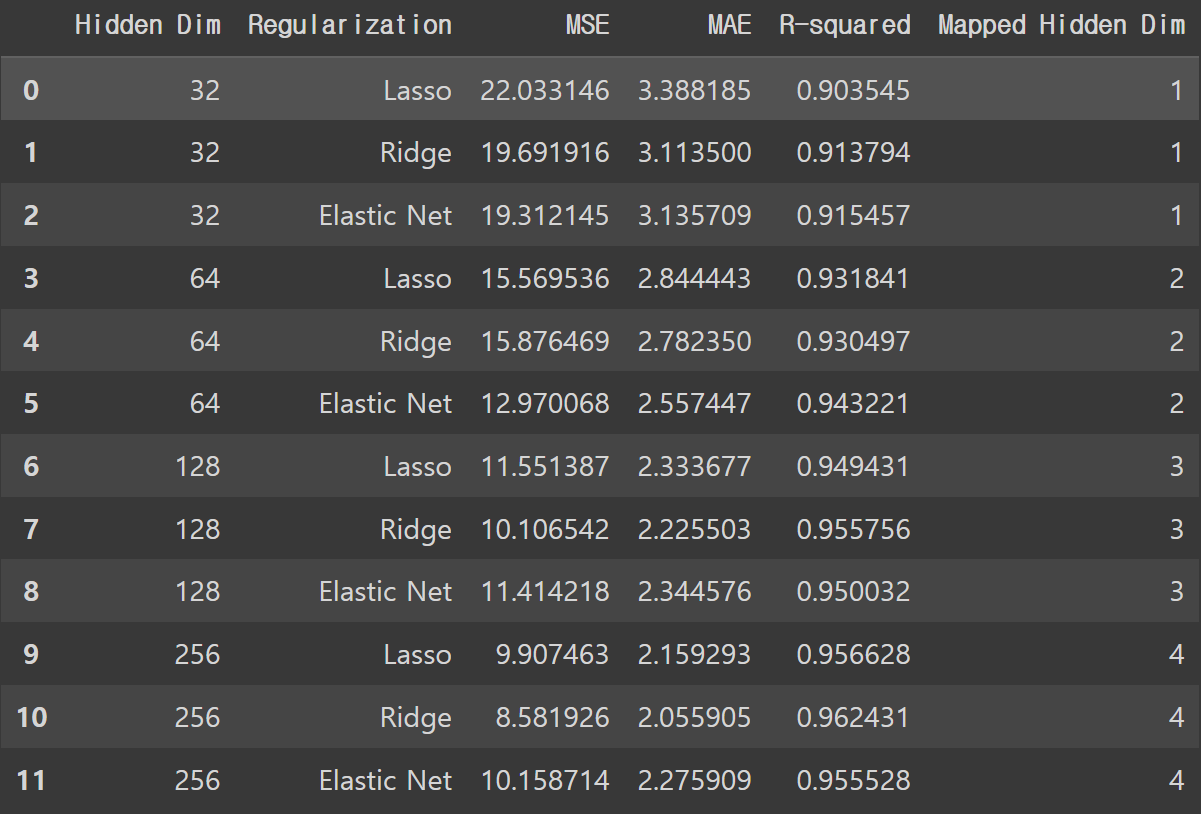
Hidden layer dimension에 따른 regularization별 MSE 변화



Hidden layer dimension에 따른 regularization별 R-squared 변화

가장 MSE가 낮고 R-squared가 높은 256 dim, L2(Ridge) Regularization의 actual data vs. predictied data 결과입니다.





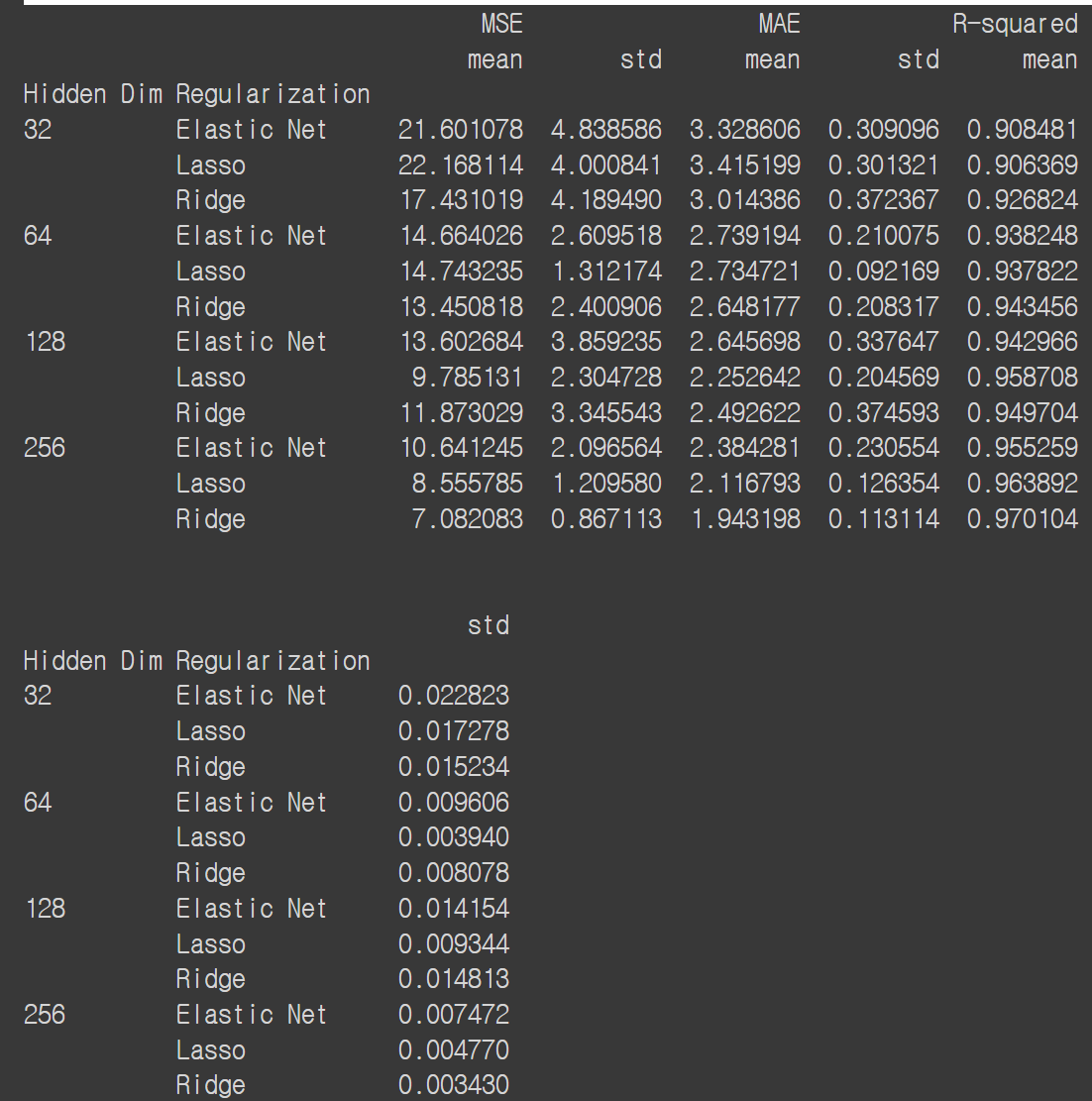
각 조건에 따른 MSE, MAE, R-squared에 대한 table입니다.

오버피팅에 대한 여부를 확인하고자 Model representation value를 구하기 위해 test data에 대해 RMSE값을 계산하였습니다. RMSE(Rooted Mean Squared Error)는 MSE에 루트를 취한 값으로 루트를 취하는 스케일링을 하였기에 이상치로부터 robust한 평가지표입니다.

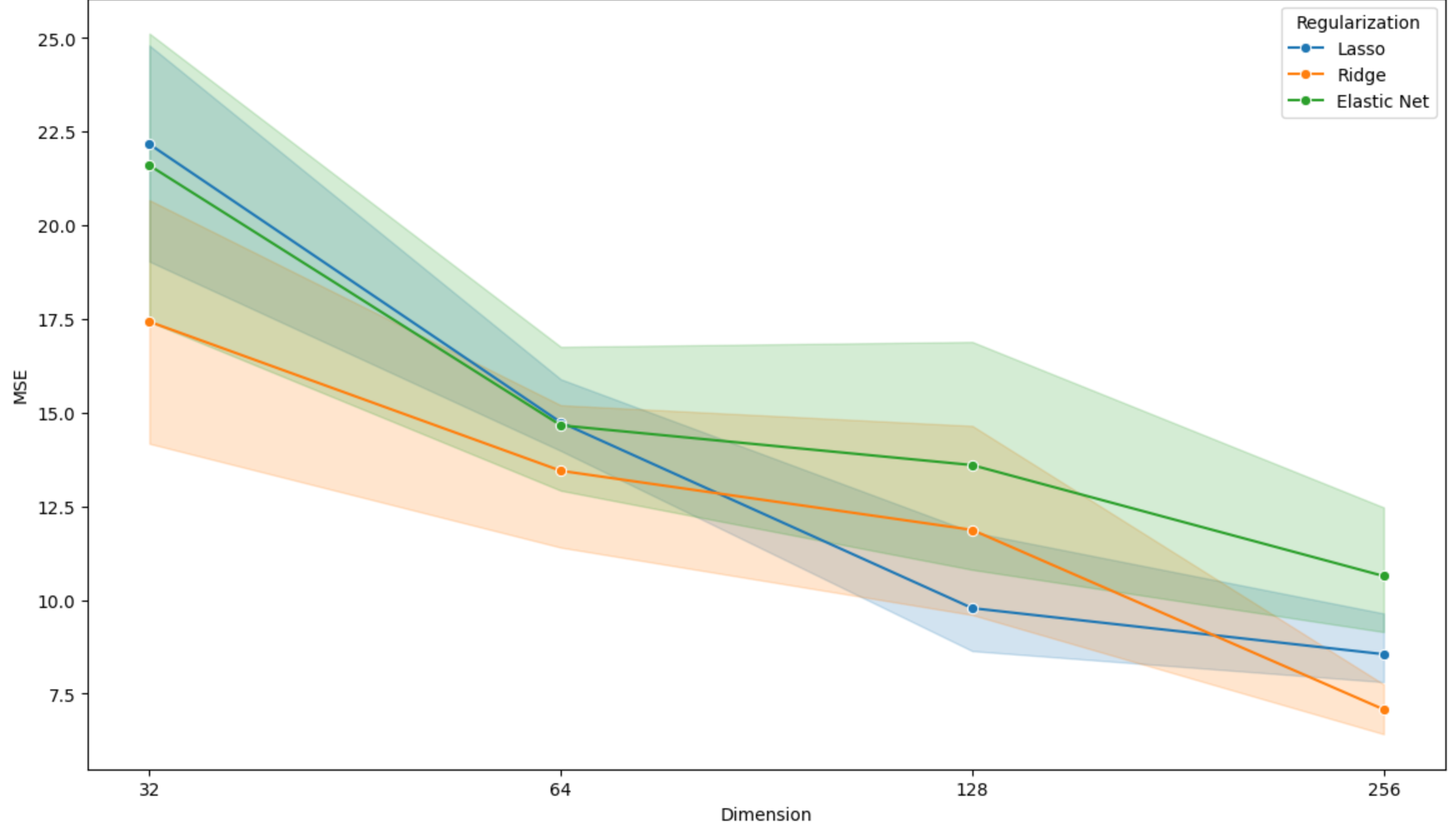
훈련에 사용된 데이터들의 예측값과 실제 값의 차이를 비교하는 MFE(Model Fit Error)가 훈련에 사용하지 않은 값들에 대한 예측값과 실제 값의 차이를 비교하는 MRE(Model Represenation Error)와 유사하다면 모델이 훈련 데이터와 테스트 데이터 모두에서 비슷한 성능을 보인다는 것을 의미합니다. 즉, 모델이 잘 일반화되어 있다고 볼 수 있습니다. Train data와 test data에 대해 RMSE 값을 계산하여 MFE와 MRE를 비교한 결과 MFE는 3.19, MRE는 3.14로 비슷한 값을 가짐을 확인할 수 있었습니다. 다시 말해 모델 적합에 있어 오버피팅이 발생하지 않았다고 볼 수 있습니다. 박유나 학우님께서 만든 코드에서 이 MRE vs. MFE를 쉽게 확인할 수 있습니다.

이 검증이 우연히 발생한 것이 아닌지를 파악하고 train-test set 구성에 따른 변동을 확인해보기 위해, K-fold cross validation을 수행해 같은 조건에서 다시 루프를 돌려보았습니다. 8:2의 train-test 분할을 유지하여 5가지 세트에 대한 각 적합도를 파악할 수 있었습니다.

이 결과를 테이블로 다시 나타내었습니다.



그래프로 CI band를 표현하면 다음과 같습니다.



여전히 Ridge가 가장 낮은 MSE를 보이며, 분산도 크지 않습니다. 반면 Elastic net의 경우 hidden layer dimension이 증가함에 따라 점점 분산이 커지는 것을 파악할 수 있습니다. 그러나 128 dimension에서는 Lasso가 Ridge보다 전반적으로 더 좋은 성능을 보이고 있습니다. 이 이유에 대해서는 각 노드의 블랙박스를 확인해보거나 소거되는 parameter가 몇 개인지 등을 확인해볼 필요가 있습니다.

결론

위도와 경도를 이용해 특정 방향의 풍속을 예측하고자 하는 모델에서, 2가지 독립변수로 종속변수 하나를 예측하는 단순한 회귀 모델로 보이지만, 두 독립변수간 interaction이 전혀 없음에도 불구하고 단순선형회귀가 불가능했습니다. 따라서 Multi-Layer Perceptron을 모델링해 2개의 hidden layer를 구성하여 적합한 모델을 만들어낼 수 있었습니다. 이 때 고정된 epoch와 activation function, optimizer 내에서 hidden layer의 dimension이 증가함에 따라 로스가 감소하는 것을 관측했고, Regularization(Ridge, Lasso, Elasticnet)을 해줌에 따라 어떤 변화가 나타나는지 피상적으로 이해할 수 있었습니다. 또한 이 모델이 오버피팅되었는지를 확인하고자 이상치로부터 안전한 RMSE를 통해 MRE와 MFE를 비교해보았고, 이후 K-fold cross validation을 활용해 5가지 train-test set에 대하여 cross validate하여 로스의 변동이 어느 정도인지 파악할 수 있었습니다. 저희 팀이 제작한 코드는 epoch와 hidden layer의 dimension, regularization, train-test 분할 등을 유연하게 조작할 수 있으며, 다른 parameter를 적용했을 때에도 쉽게 성능을 평가할 수 있습니다.