Machine Learning in Finance - Homework 3 – Due 09.24.2024

Textbook reading:
Chapters 4

- **Upload Regularization.ipynb to colab**
- **To remove the error message when running the Lasso Regression, replace Y_train with y_train.ravel(). What does the ravel function do? Does this change the answer?**
  The numpy.ravel() function flattens an array into 1D shape. The error message appears because of the creation of $Y = X \cdot \beta + \epsilon$ returns a 2D array by Numpy's standard. No, the ravel function does not change the answer. Python uses broadcasting to treat the array.

- **For the Lasso regression**
  - **What does the argument cv=20 do?**
    It sets the cross-validation parameter to 20 folds to come up with the best hyperparameter.

  - **What is the penalty value selected by the algorithm?**
    The penalty value, commonly known as lambda in $\lambda\sum|x|_1$ can be found by using the command .alpha_. In this case, it is 0.015597830979571946.

  - **Out of the 800 fields, how many had non-zero coefficients?**
    42.

  - **How many non-zero coefficients are computed when alpha=0.09? How does this change the look of the plot Y-observed vs Y-predicted?**
    Only 3. By setting alpha = 0.09 when the optimal alpha is approximately 0.016, the program intentionally leads the model to higher bias. The relatively high penalty value had led to the model trying to minimize the l_1 norm instead of the MSE of the linear model. This resulted in forcing many of the coefficients to 0. The plot Y-observed vs Y-predicted pushed the scatter dots off the main diagonal, which is a result of higher bias.

- **For the Ridge regression**
  - **What was the computed penalty function?**
    10.

  - **How many non-zero coefficients were there?**
    800. Nothing was dropped.

  - **How does the distribution of coefficients change if alpha=0.5?**
    Increased alpha leads to driving more coefficients towards 0 but not exactly 0. The effect on the distribution is then typically: the mean decreases, standard deviations

decrease, skewness increases (more asymmetric as most values are concentrated around the mean), kurtosis increases (more pointy pdf).

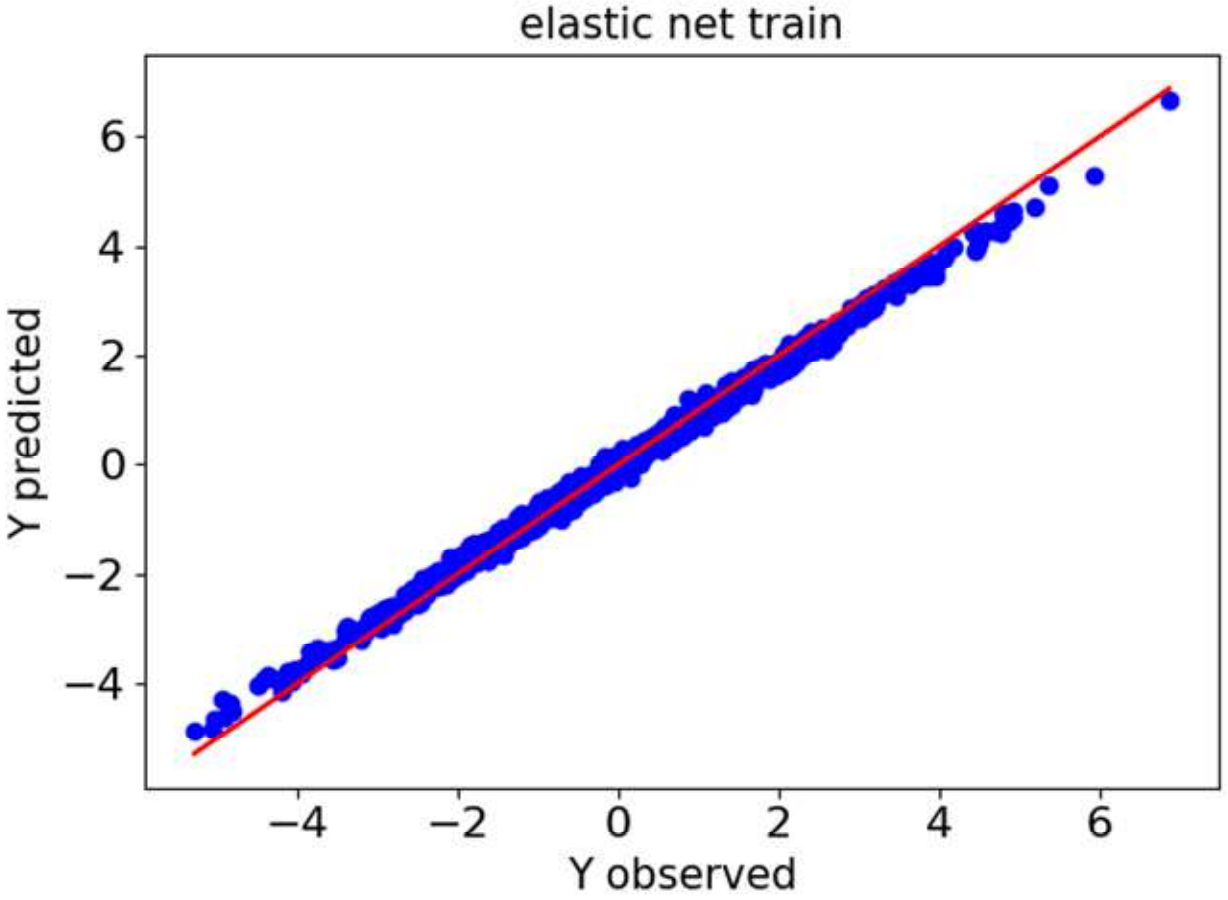
- **From the rmse and r2 statistics for test and training which model performs best? Explain your conclusion in terms of over- or under-fitting**
  Lasso regression model performs the best in terms of test statistics. Ridge regression has better training statistics. This is because it attempts to fit the noise from the training data. As such it overfits. Linear regression performs poorly in both training and test, because it underfits.

- **Compute Elastic Net solution**

```
from sklearn.linear_model import ElasticNetCV

elasticnet = ElasticNetCV(cv=5, random_state=0)
elasticnet.fit(X_train, Y_train)
print("alpha is ", elasticnet.alpha_)
print("l1 is ", elasticnet.l1_ratio_)
elasticnet_train = display_predictions(elasticnet, X_train, Y_actual_train, Y_train, "elastic
net", "train", 'ElasticNet_train.png')

alpha is  0.02359837425088335
l1 is  0.5
ElasticNet_train.png -  training set size = (960,)
r^2 on elastic net train data : 0.944385
rmse on elastic net train data : 0.472082
```




```
elsticnet_test = display_predictions(elasticnet, X_test, Y_actual_test, Y_test, "elastic net",
"test", 'Elastic_test.png')

Elastic_test.png -  training set size = (960,)
r^2 on elastic net test data : 0.937332
rmse on elastic net test data : 0.538460
```

To obtain documentation on lassocv, search for sklean and lassocv