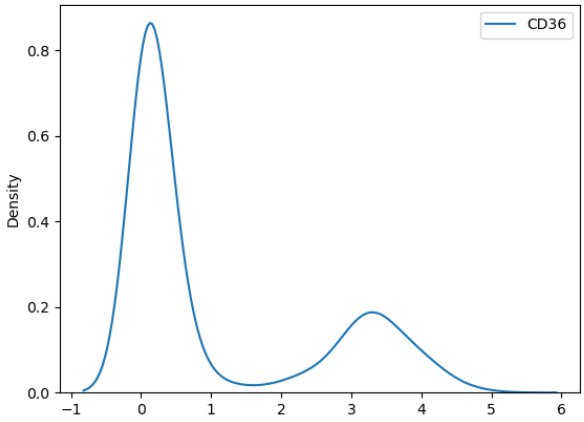
1(a)
X_train has 4302 observations and 6936 variables.
Data is of type float64, and is complete.

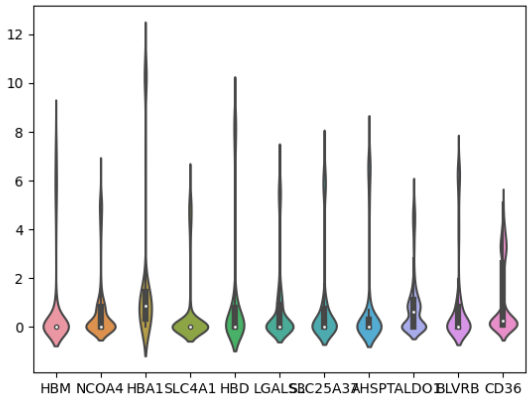
X_test has 761 observations and 6936 variables.
Data is of type float64, and is complete.

Y_train has 4302 observations.
Data is of type float64, and is complete.

1(b)
Distribution of Y_train looks like mixture of
2 normal distributions one with peak at 0.2,
other with peak at 3.3

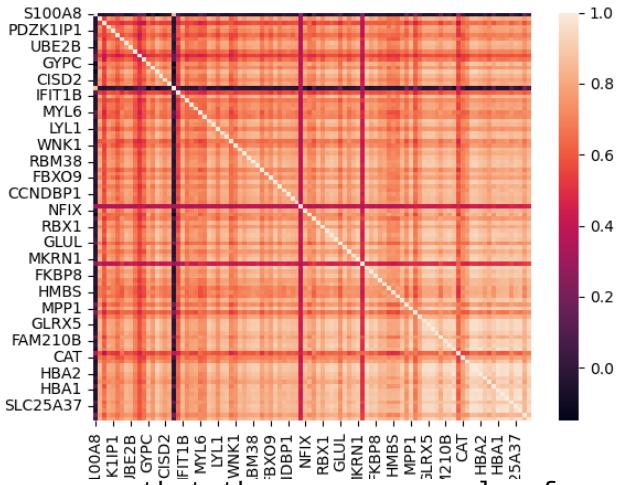


1(c)
violin plot of top 10 variables most correlated to
response variable.



Some of them have quite similar distributions to
response variable.
Not very surprising since there is a lot of
variables in the dataset.

Heatmap of correlations between top 100 variables
most correlated to response variable.



We can see that there are a couple of variables
not correlated to any others.
There is also a lot of pairs of highly correlated
variables, which makes sense since we selected
variables most correlated to response variable.

2(a)
ElasticNet model is a linear regression model
that combines penalties of lasso and ridge regression.

Optimization function is 1 / (2 * n_samples) * ||y - Xw||^2_2
+ alpha * l1_ratio * ||w||_1
+ 0.5 * alpha * (1 - l1_ratio) * ||w||^2_2

Hyperparameters include
alpha - constant that multiplies penalty terms.
l1_ratio - ratio of lasso penalty to total penalty.
If l1_ratio = 1 penalty is lasso penalty.
If l1_ratio = o penalty is ridge penalty.

2(b)
Before training I did PCA decomposition mostly to speed up training.
I think its justified, since there are highly correlated variables.
Most important hyperparameters for ElasticNet are alpha and l1_ratio.
I chose {'alpha':[0.5, 1, 2], 'l1_ratio':[0, 0.5, 1]}
as a grid of hyperparameters. And chose 5-fold cross validation
because 800 instances is enough to test the model,
and its also about the size of test set.

2(c)
Best hyperparameters were {'alpha': 2, 'l1_ratio': 0}
with 0.41378538867190573 RMSE. (Without PCA RMSE was about 0.38)

3(a)
Most important features of random forest are number of trees and their size.
So for hyperparameters I chose n_estimators, max_depth.
Additionaly i messed with criterion hyperparameter.

3(b)
Random forest acheived 0.35398594232973635 RMSE with
{'criterion': 'squared_error', 'max_depth': 10, 'n_estimators': 20}
as hyperparameters. GridSearchCV gives same splits across calls so
Random forest is a better model.
For reference model assigning arithmetic mean achieves
1.4599745884698907 RMSE.

4
For prediction on the test set i used Random forest after PCA
with hyperparameters tuned using 5-fold cv.