Big Data Asset Pricing

Exercise 4: Hig-Dimensional Return Predictions

Seyyed Morteza Aghajanzadeh

Department of Finance

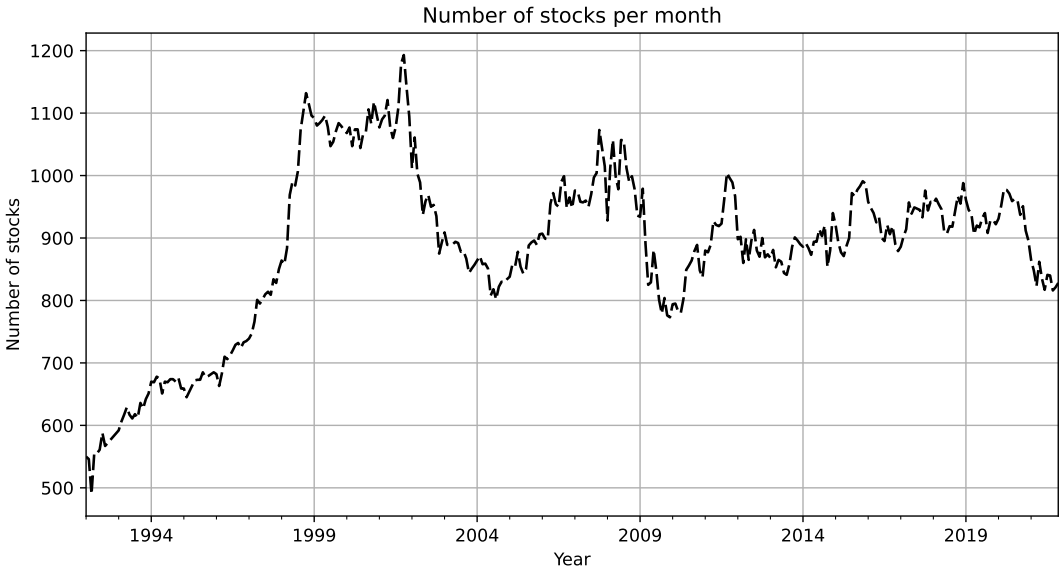
Stockholm School of Economics

Statement: I certify with my signature that I have solved the exercise according to the Code of Professional Conduct and Ethics. For example, I have not plagiarized others, but, instead, solved the exercise myself (possibly with allowed collaboration with other students), and I have referenced my sources appropriately.

2024-01

1

Figure 1: The number of stocks per month.



2

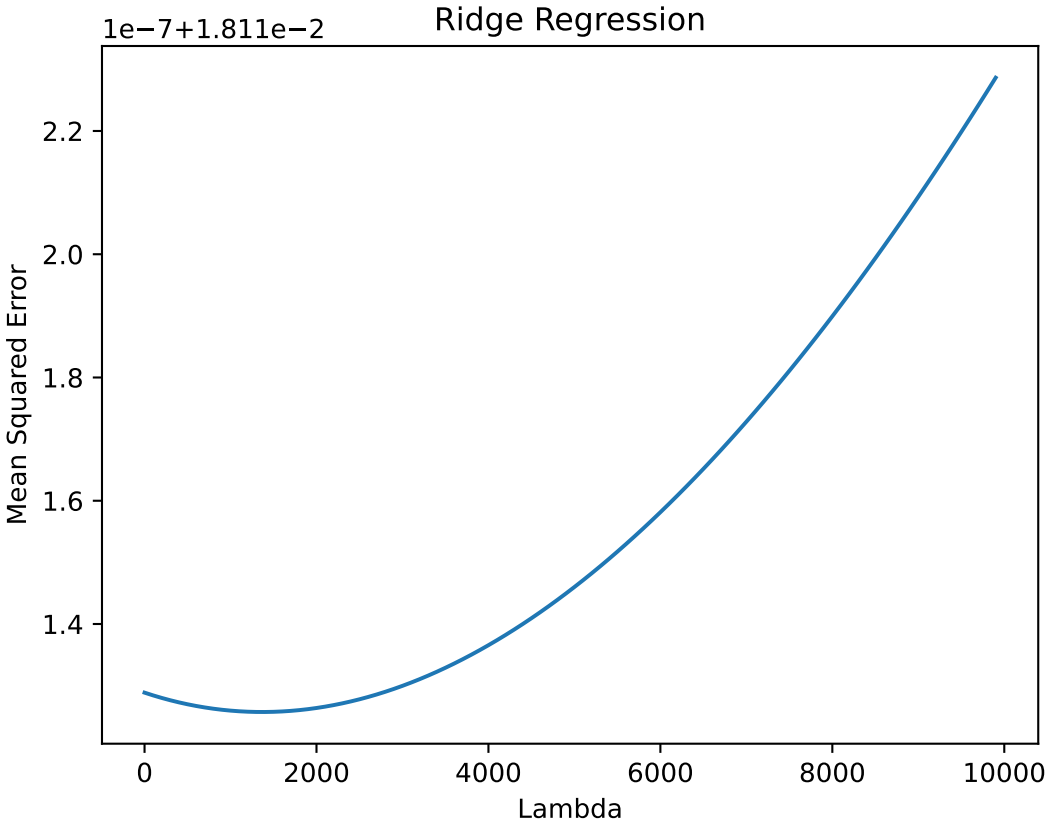
2.a

Table 1: The results of the Pooled OLS regression over training

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
be_me	0.0316	0.0263	1.2032	0.2289	-0.0199	0.0831
ret_12_1	-0.0239	0.0211	-1.1340	0.2568	-0.0652	0.0174
market_equity	-0.0316	0.0276	-1.1424	0.2533	-0.0858	0.0226

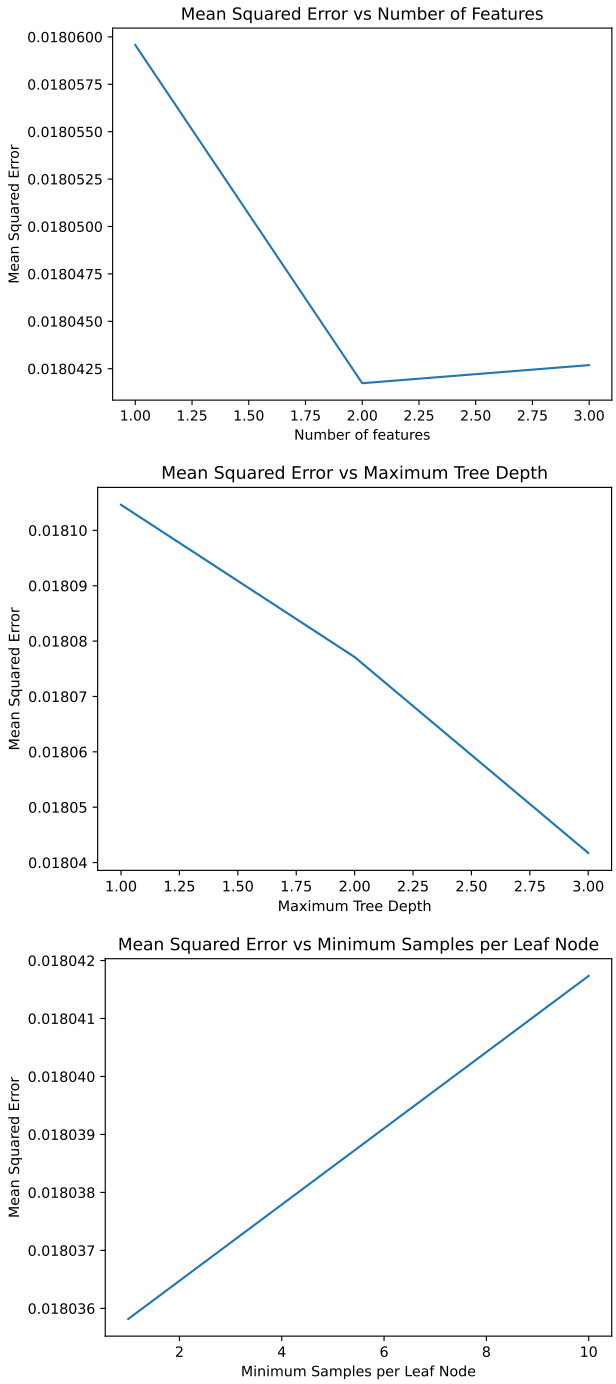
2.b

Figure 2: The mean squared error on the validation set, as a function of λ.



2.c

Figure 3: The mean squared error on the validation set, as a function of different features.



3

The results show that both the Pooled OLS (Model A) and Ridge regression (Model B) have the same in-sample R-squared, indicating that they explain a similar proportion of the variance in the data. However, the Random Forest model has a higher in-sample R-squared, suggesting that it provides a better fit to the data.

A higher in-sample R-squared indicates that the Random Forest model captures more of the underlying patterns and relationships in the data, leading to a better prediction performance. This could be attributed to the Random Forest's ability to handle non-linear relationships and interactions between variables, which may not be adequately captured by the linear models like Pooled OLS and Ridge regression.

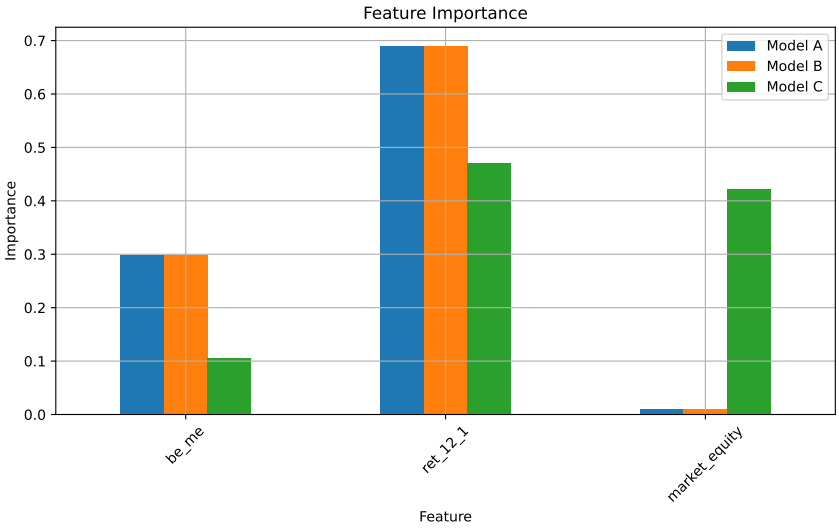
It is important to note that while a higher in-sample R-squared suggests better performance within the training data, it does not necessarily guarantee better out-of-sample prediction accuracy. Therefore, it is crucial to evaluate the models' performance on unseen data to assess their generalization ability.

Table 2: The In-sample R-squared for the Pooled OLS, Ridge Regression, and Random Forest.

Model	In-sample R2
Model A	0.0012
Model B	0.0012
Model C	0.0050

4

In this analysis, we explore the feature importance of each model: Pooled OLS, Ridge regression, and Random Forest. While both Pooled OLS and Ridge regression assign equal importance to the features, they both highlight the significance of the lagged return as the most influential factor. This aligns with existing literature, which suggests that the 12-month momentum serves as a robust predictor of future returns. However, where the models diverge is in the importance they assign to other features.



5

5.a

Here we compare the out-of-sample performance of the Pooled OLS, Ridge regression, and Random Forest models. The results show that the Pooled OLS outperform the Ridge regression and Random Forest in terms of out-of-sample R-squared. This suggests that the Pooled OLS model provides a better fit to the data and captures more of the underlying patterns and relationships, leading to better prediction performance.

Table 3: Out-of-sample R-squared for the Pooled OLS, Ridge Regression, and Random Forest.

Model	Out-sample R2
Model A	0.4869
Model B	-0.0007
Model C	0.0002

5.b

I followed the instruction and create 5 portfolios based on the predicted returns for each model. The results show that the Ridge regression outperforms the Pooled OLS and Random Forest in terms of the Sharpe ratio. This suggests that the

Ridge regression model provides a better risk-adjusted return, which is an important measure of investment performance.

Table 4: Portfolio performance based on the predicted returns for the Pooled OLS, Ridge Regression, and Random Forest.

Panel A: Pooled OLS

Portfolio	$r_i - r_f$	t_{stat}	α_{CAPM}	t_α	Sharpe Ratio	IR
1	0.008	1.621	0.008	1.472	0.149	0.139
2	0.008	1.551	0.008	1.447	0.142	0.134
3	0.006	1.160	0.006	1.146	0.106	0.105
4	0.009	1.798	0.009	1.767	0.165	0.162
5	0.003	0.825	0.003	0.938	0.076	0.082
LS	-0.005	-1.515	-0.004	-1.200	-0.139	-0.118

Panel B: Ridge Regression

Portfolio	$r_i - r_f$	t_{stat}	α_{CAPM}	t_α	Sharpe Ratio	IR
1	-0.481	-0.787	-0.526	-0.816	-0.072	-0.079
2	0.061	0.802	0.075	0.885	0.074	0.092
3	-0.010	-0.168	-0.012	-0.226	-0.015	-0.019
4	0.003	0.071	0.003	0.080	0.007	0.006
5	0.060	1.770	0.055	1.848	0.162	0.149
LS	0.540	0.884	0.580	0.900	0.081	0.087

Panel C: Random Forest

Portfolio	$r_i - r_f$	t_{stat}	α_{CAPM}	t_α	Sharpe Ratio	IR
1	-0.068	-0.380	-0.074	-0.413	-0.035	-0.038
2	-0.019	-1.221	-0.020	-1.214	-0.112	-0.120
3	-0.059	-0.745	-0.032	-0.510	-0.068	-0.038
4	0.008	1.676	0.008	1.605	0.154	0.147
5	0.013	2.508	0.012	2.334	0.230	0.222
LS	0.080	0.452	0.086	0.485	0.041	0.045

6

Now I use all the mentioned features to predict the returns. Unfortunately, my results for the Random forest model do not look reasonable and I could not fix the

problem. I presented the results here following the previous structure. But I cannot discuss these results as they are not reliable.

6.2

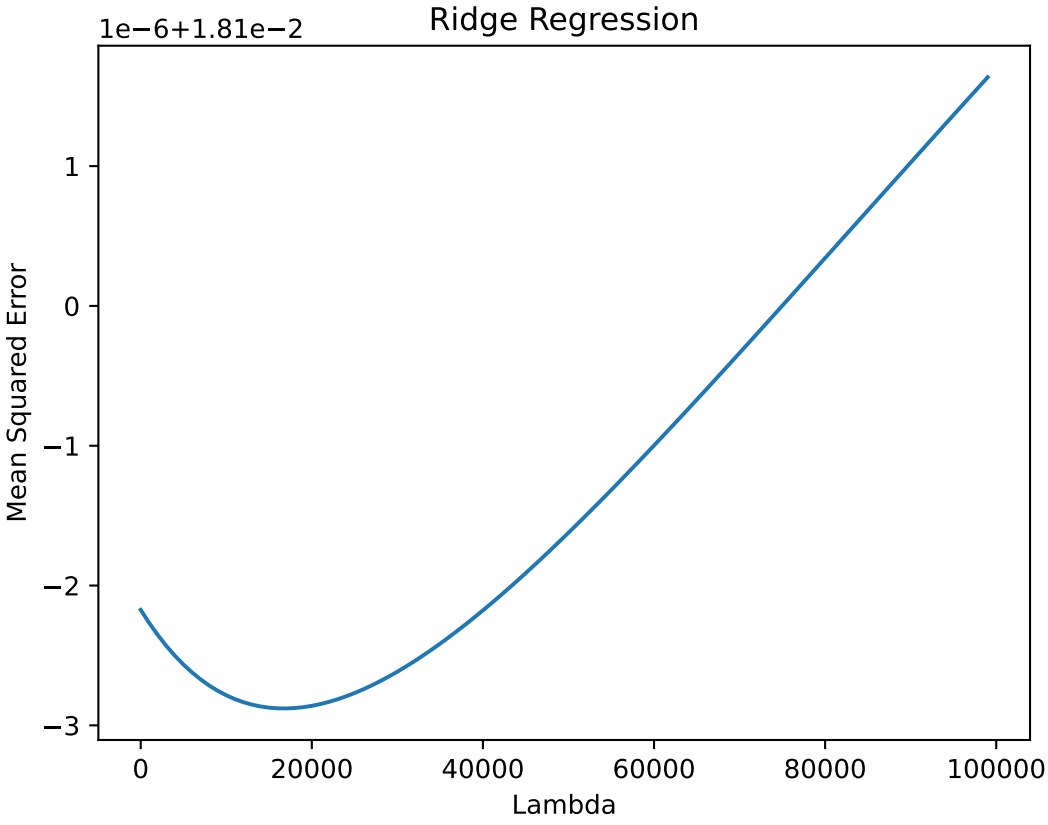
6.2.a

Table 5: The results of the Pooled OLS regression.

	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
be_me	0.0041	0.0128	0.3203	0.7487	-0.0210	0.0292
ret_12_1	0.0042	0.0048	0.8691	0.3848	-0.0053	0.0137
market_equity	0.1608	0.1310	1.2274	0.2197	-0.0959	0.4175
ret_1_0	0.0012	0.0048	0.2500	0.8026	-0.0081	0.0105
rvol_252d	0.0001	0.0109	0.0116	0.9907	-0.0213	0.0215
beta_252d	-0.0947	0.0857	-1.1057	0.2688	-0.2627	0.0732
qmj_safety	0.0068	0.0035	1.9140	0.0556	-0.0002	0.0137
rmax1_21d	-0.0931	0.1244	-0.7484	0.4542	-0.3370	0.1507
chcsho_12m	-0.0089	0.0621	-0.1438	0.8856	-0.1307	0.1129
ni_me	-0.0562	0.0472	-1.1901	0.2340	-0.1486	0.0363
eq_dur	43.781	43.188	1.0137	0.3107	-40.865	128.43
ret_60_12	-0.0098	0.0042	-2.3206	0.0203	-0.0180	-0.0015
ope_be	-0.0192	0.0309	-0.6206	0.5349	-0.0798	0.0414
gp_at	0.0024	0.0029	0.8322	0.4053	-0.0033	0.0081
ebit_sale	-0.6719	0.6837	-0.9827	0.3258	-2.0119	0.6682
at_gr1	-1.0733	0.7334	-1.4635	0.1433	-2.5107	0.3641
sale_gr1	0.2397	0.2739	0.8753	0.3814	-0.2971	0.7765
at_be	0.0777	0.0427	1.8208	0.0686	-0.0059	0.1613
cash_at	-0.0002	0.0030	-0.0562	0.9552	-0.0060	0.0057
age	9.9438	7.5646	1.3145	0.1887	-4.8827	24.770
z_score	0.0423	0.0594	0.7122	0.4764	-0.0741	0.1588

6.2.b

Figure 4: The mean squared error on the validation set, as a function of λ.



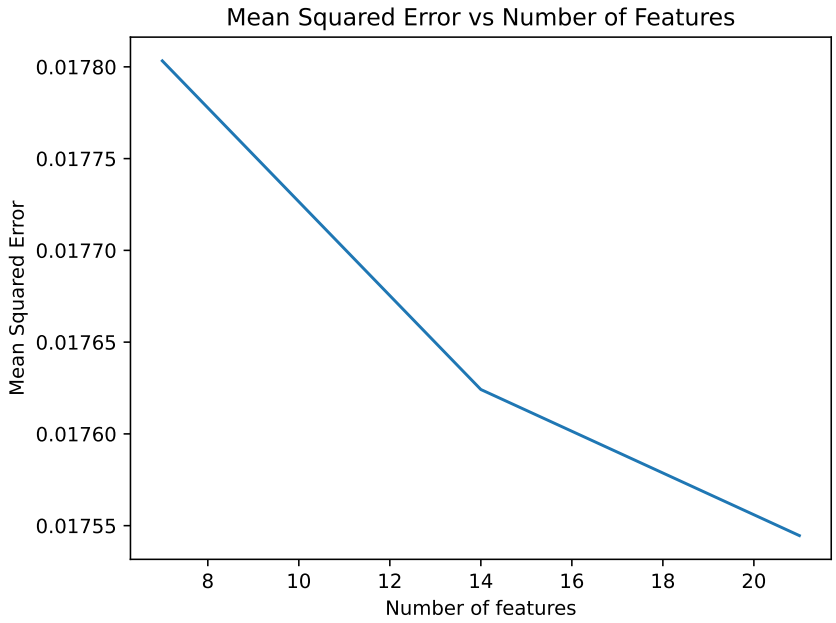
6.2.c

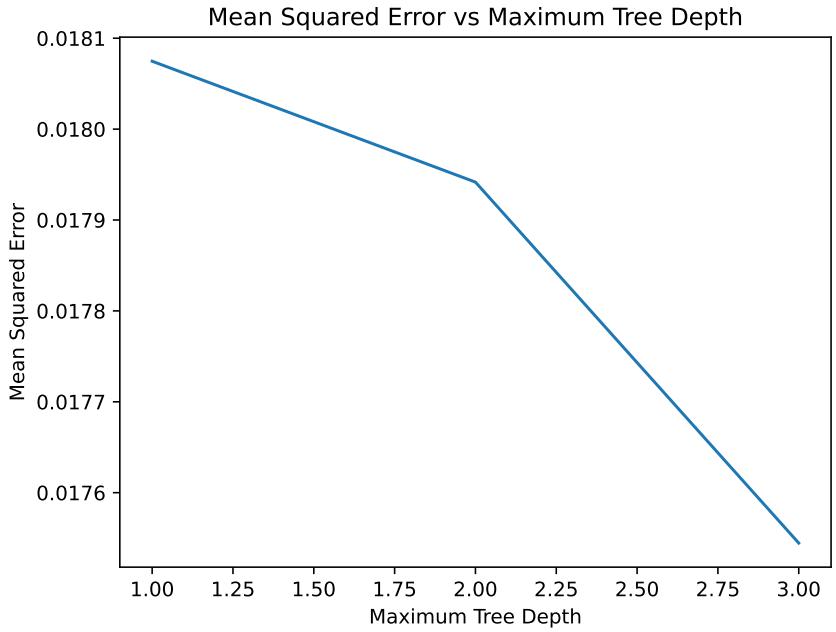
6.3

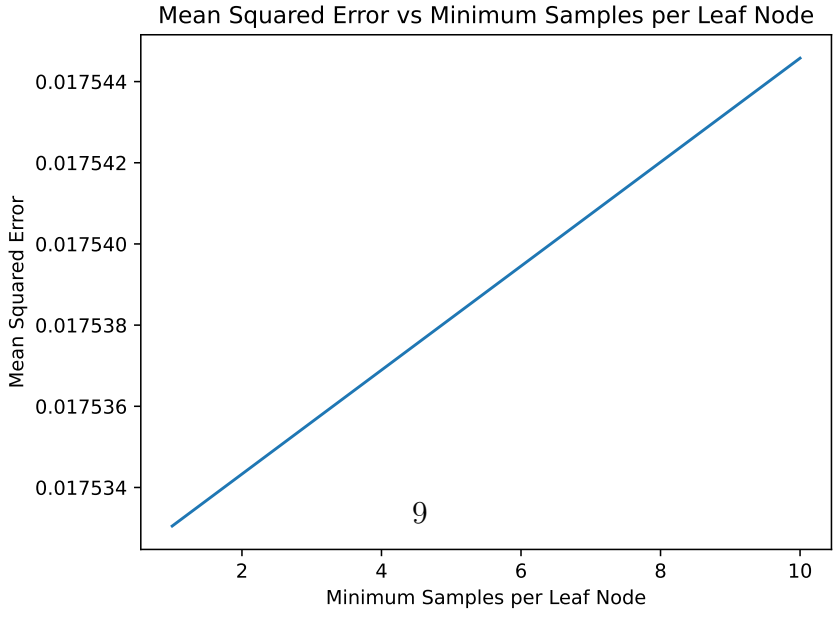
Table 6: The In-sample R-squared for the Pooled OLS, Ridge Regression, and Random Forest.

Model	In-sample R2
Model A	0.0023
Model B	0.0023
Model C	0.0324

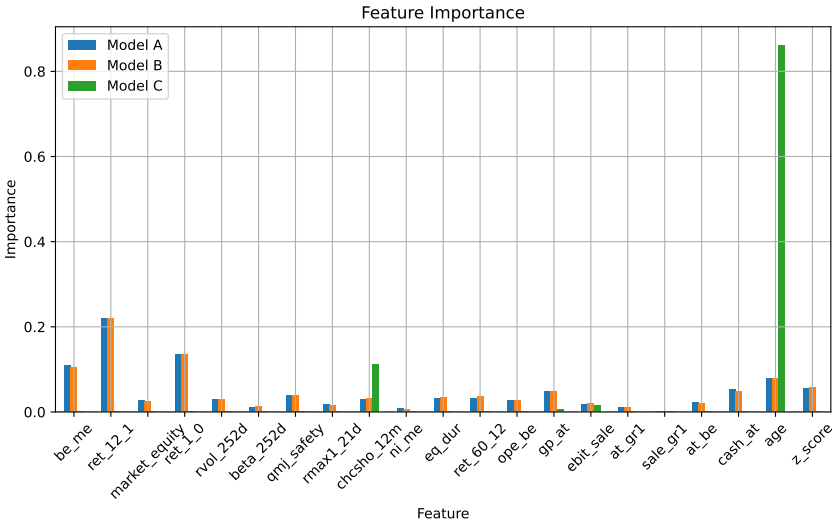
Figure 5: The mean squared error on the validation set, as a function of different features.







7



8

8.1

Table 7: Out-of-sample R-squared for the Pooled OLS, Ridge Regression, and Random Forest.

Model	Out-sample R2
Model A	0.4870
Model B	0.0003
Model C	-0.0298

8.2

Table 8: Portfolio performance based on the predicted returns for the Pooled OLS, Ridge Regression, and Random Forest.

Panel A: Pooled OLS

Portfolio	$r_i - r_f$	t_{stat}	α_{CAPM}	t_α	Sharpe Ratio	IR
1	0.037	2.011	0.041	2.036	0.184	0.209
2	-0.001	-0.050	0.001	0.085	-0.005	0.005
3	0.032	0.167	0.042	0.207	0.015	0.020
4	0.014	1.276	0.014	1.341	0.117	0.121
5	0.016	2.844	0.016	2.599	0.261	0.252
LS	-0.020	-1.023	-0.025	-1.174	-0.094	-0.118

Panel B: Ridge Regression

Portfolio	$r_i - r_f$	t_{stat}	α_{CAPM}	t_α	Sharpe Ratio	IR
1	0.129	1.311	0.128	1.307	0.120	0.120
2	-0.005	-0.419	-0.004	-0.359	-0.038	-0.032
3	-0.030	-0.556	-0.031	-0.554	-0.051	-0.052
4	0.011	0.963	0.008	0.698	0.088	0.069
5	0.017	2.882	0.016	2.600	0.264	0.255
LS	-0.112	-1.155	-0.113	-1.170	-0.106	-0.106

Panel C: Random Forest

Portfolio	$r_i - r_f$	t_{stat}	α_{CAPM}	t_α	Sharpe Ratio	IR
1	-0.061	-0.995	-0.032	-432593149695673.500	-0.091	-3264846760324042.000
2	0.402	0.785	0.142	586305580512332.625	0.072	inf
3	0.115	1.281	-15.369	-2109578597928374.750	0.117	-12837351900248766.000
4	0.061	7.593	0.151	987705348430798.125	0.696	6845086170022979.000
5	-0.073	-34.301	0.018	137925016638561.375	-3.144	1847950789082325.000
LS	-0.034	-17.384	0.050	2489526277370856.500	-1.594	inf

Appendix

Here you can find the python code that I used to solve the exercise. Link to the GitHub repository.