PRIVATE EQUITY

NOWCASTING

Results & Recommendations

Executive Summary:

The private equity market has been taking more relevant role on many large portfolios, however the private equity investors don't have access to the same level of information compared to public equity investors. This shortcoming is also applied to our client AFP Habitat. However, we believe that gap can be minimized through the use of Bayesian models as well as other machine learning techniques including advanced time series and feature selection. In this project called Private Equity Nowcasting, public data points in conjunction with internal information will be utilized to better understand the return of private assets and gain insights into the private market regimes. Our aim is to boost the competitive analytics edge at AFP Habitat.

Project Problems and Objectives

There are two main concerns that AFP Habitat is facing:

- The CAPM model has limitations and simply is unsatisfactory for today's prediction. Our team will explore, test different types of predictive models and construct one that yields better results on return analysis.
- There are still uncertainties regarding the relationship between public market factors and returns on various assets. We will investigate the influential components of public data points on AFP private assets performance.

With this in mind, the primary objective of the project is testing and developing models to estimate the performance of chosen assets, based on public market

indicators. In addition, our secondary goals are: identifying Bayesian confidence interval, updating private market's regimes, and examining the sensibility of the chosen model.

Project Execution Plan

1. Data Cleaning
2. Feature Selection
3. CAPM Model
4. LSTM Model
5. CNN Model
6. Bayesian Model

1. Data Cleaning

Programming Language: R

We receive 3 data files that have different structure and recording period. However, we are able to combine all of them to 1 data file with over 400 variables between a select time period of 2000 till 2018. To treat missing values by dates, we impute them by previous data point consistently. In which result in over 6,000 observations for daily and about 200 for monthly.

2. Feature Selection

Programming Language: R

To diversify our analysis, 5 important assets from different types are chosen for feature selection analysis:

a. Private asset
b. Real Estate
c. Buyout
d. Venture
e. Private Capital - Euro

The objective is to find predictors that are associated with the outcome and are not related to other predictors. To do so, we run a simple regression to find the statistical significant then we choose predictors with highest coefficients, the threshold is p-value is smaller than .001. After that, we inspect threat of multicollinearity to ensure all independent variables are not related to each other. We compute Variance Inflating Factor and the benchmark value for VIF is 5. Last but not least, we test all possible subset with hybrid stepwise, shrinking method, and regssubset with lowest cp. The results yield important macro indices that have an impact on 5 chosen AFP's assets. For instance, slope Chile has an inverse relationship with asset Private Equity, while Atlanta Fed GDP Nowcast has a positive correlation and so on. (see next page)

Macro Indices	Private Equity	Real Estate	Buyout	Venture	Private Capital-Europe
Argentina Price To Earningsn Ratio			x	x	
Atlanta Fed GDP Nowcast (2011)	x	(x)	x		x
Brazil LC Price To Earnings Ratio	(x)		(x)	x	
Europe Small Cap Price To Earnings Ratio		(x)			
Germany Price To Earnings Ratio		x			
India Total Return Index Gross Dividends					x
Japan Large Cap Price To Earnings Ratio				x	(x)
Japan Small Cap Price To Earnings Ratio		x			
Korea Price To Earnings Ratio	x		x		x
Private Equity Index		x			
Slope Chile	(x)				
USA Large Cap Blend DVD_SH_12M	x		x		
USA Large Cap Growth Closing Price				x	
USA Small Cap Growth Price To Earnings Ratio	x		x	x	
USA Small Cap Value DVD_SH_12M		x			

Feature Selection Results

x: positive correlation

(x): negative correlation

3. CAPM Model

Programming Language: R

The very first model we built is the CAPM model. It serves as a benchmark for other models that we built later. For the CAPM model, we make 2 assumptions at the beginning:

- First, we presume "Slope.CL" as the Risk-Free Rate for asset named Private Equity, Real Estate, Buyout, Venture. For asset named "Private capital-Europe", we presume "Slope.EU" as the risk-free rate since it's the specific asset in the European area.
- Second, we utilized variables based on the feature selection to avoid issues such as model overfitting.

CAPM Model Results

Model	***Data Usage***	***Language***	***TEST RMSE***				
			Private Equity	***Real Estate***	***Buyout***	***Venture***	***Private Capital-Europe***
CAPM	*Daily*	*R*	*22.4*	*29.01*	*33.54*	*8.78*	*41.62*

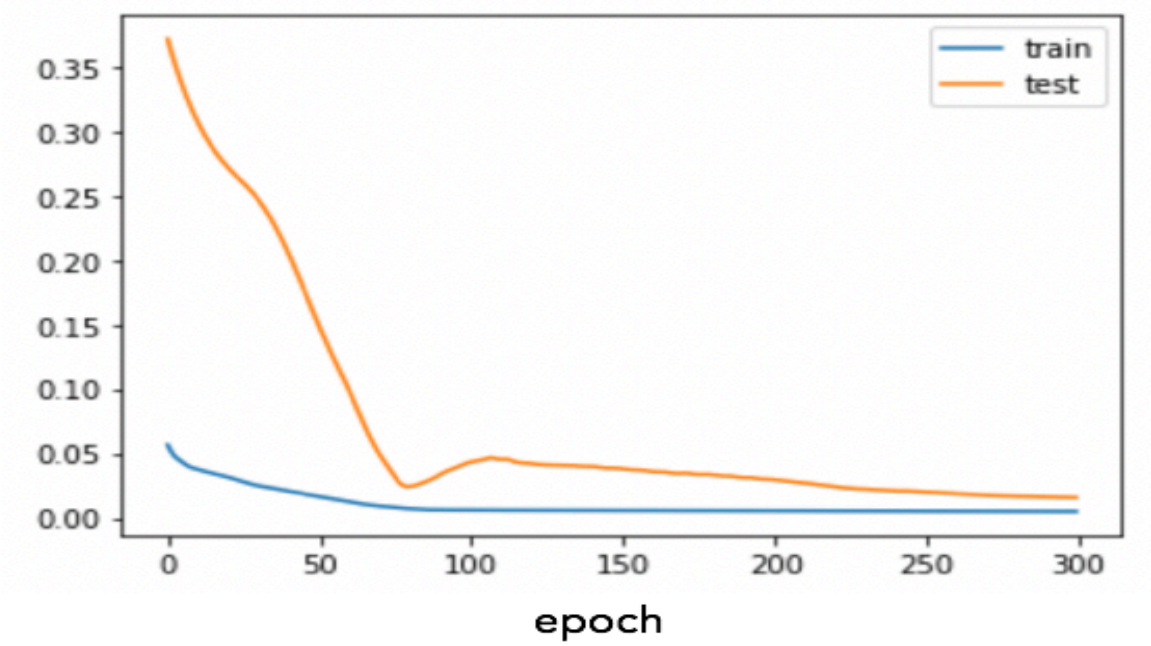
While the CAPM model is widely used, the primary drawbacks are reflected in the model's inputs and assumptions, including: Risk-Free Rate, Return on the Market, and Ability to Borrow at a Risk-Free Rate. In other words, the results are biased based on input assumptions. With this in mind, we find that within the CAPM model there is inconsistency with testing results.

4. LSTM Model

Programming Language: Python

Long short-term memory (LSTM) can process single data points, but also sequences of dataset. With this in mind, the model is chosen to classify, process and make predictions based on time series dataset. The first step is to prepare the dataset by splitting it into train and test with 2:1 ration, however the initial results are not ideal, so the splitting ration is changed to 5:1. After that, train and test sets are split into inputs and outputs variables. Then, to meet the format expected by LSTM, they are reshaped into 3D format. Next step is to design and fit the LSTM model: 50 neurons in the first hidden layer and 1 neuron in the output layer for predicting returns. The mean absolute error loss function and Adam optimizer are used to compile the model.

To fit the model, 50 training epochs with a batch size of 72 at first are set. Moreover, to make the model performs better, epochs number is changed to 300 to increase the hidden layers and batchsize is altered to 1200 to multiply the number of training examples in one forward/backward pass. After that, we tracked the value loss for train and test in each epoch by visualization (see below).

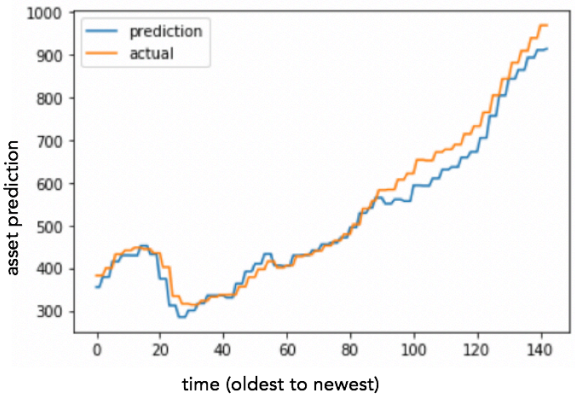


From this plot, the value losses keep decreasing, which is a good sign. When the model is fit, the entire test set is forecasted. We combine the forecast with the test set and invert the scaling with the expected returns.

LSTM Model Results

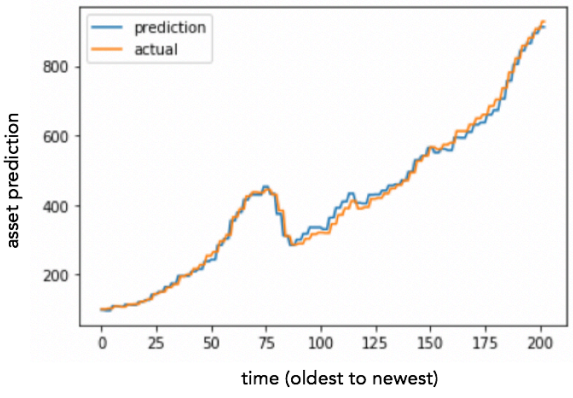
With forecasts and actual values in their original scale, we calculate the RMSE for the model using quarterly dataset which was 16.79 and for monthly which was

Monthly Data



- Train RMSE: **61.99**
- Test RMSE: **32.18**

Quarterly Data



- Train RMSE: **5.91**
- Test RMSE: **16.79**

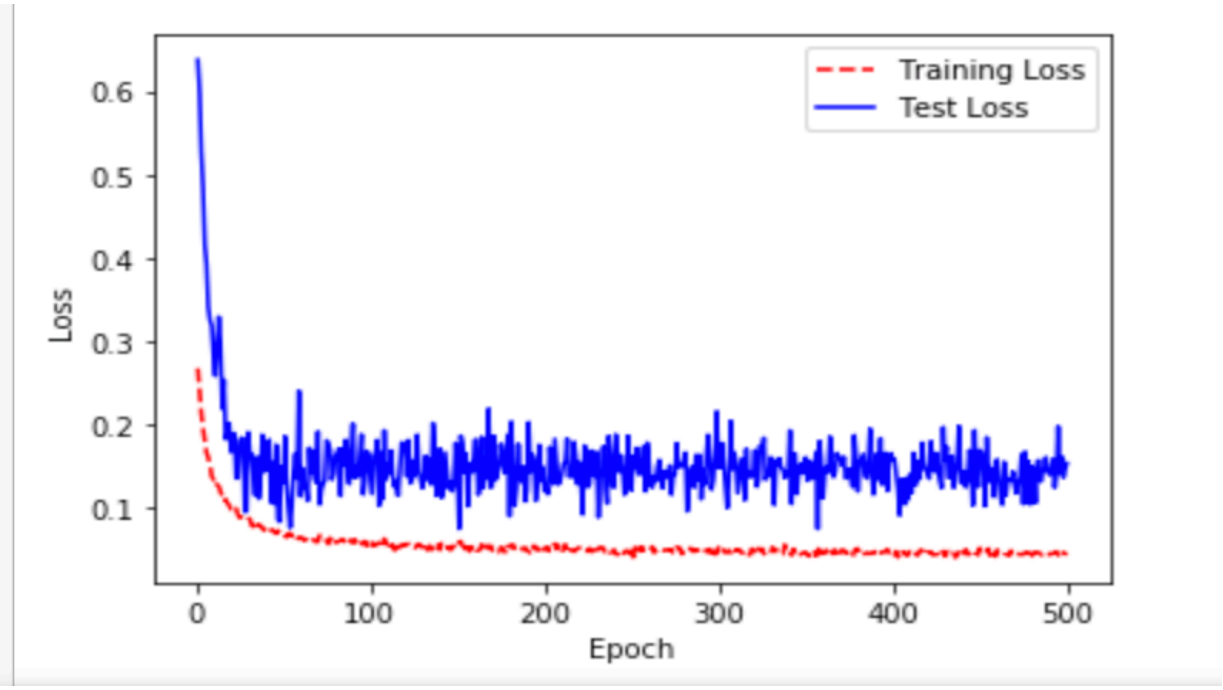
32.18. These two plots above are actual data and our predictions. We could see the LSTM model did well with test set, but it seems to be overfitting because of the big gap between RMSE of train and test set.

5. CNN Model

Programming Language: Python

CNN has widely used in action recognition, medical imaging and computer vision. There are 3 types of CNN model which are 3D, 2D, and 1D. However, 1D CNN is chosen due to the fact that this is a pure times series case. Moving to the model construction part, there are three hidden layers: convolutional layers, pooling layers and fully connected layers. Generally, the more combo of convolutional layers and pooling layers a model has, the deeper analysis it can go, and our final version of CNN has two combos. Then we have two options to train our model: one is sliding windows, and the other is normal training. We did not choose sliding window because the test loss in sliding window methodology is upwarding and the reason may be the monthly dataset is too small.

The CNN model includes the five chosen assets, with a train-test split ratio of 0.8. It's a time series model without any macro variable included, because variable selection is not applicable in this model.



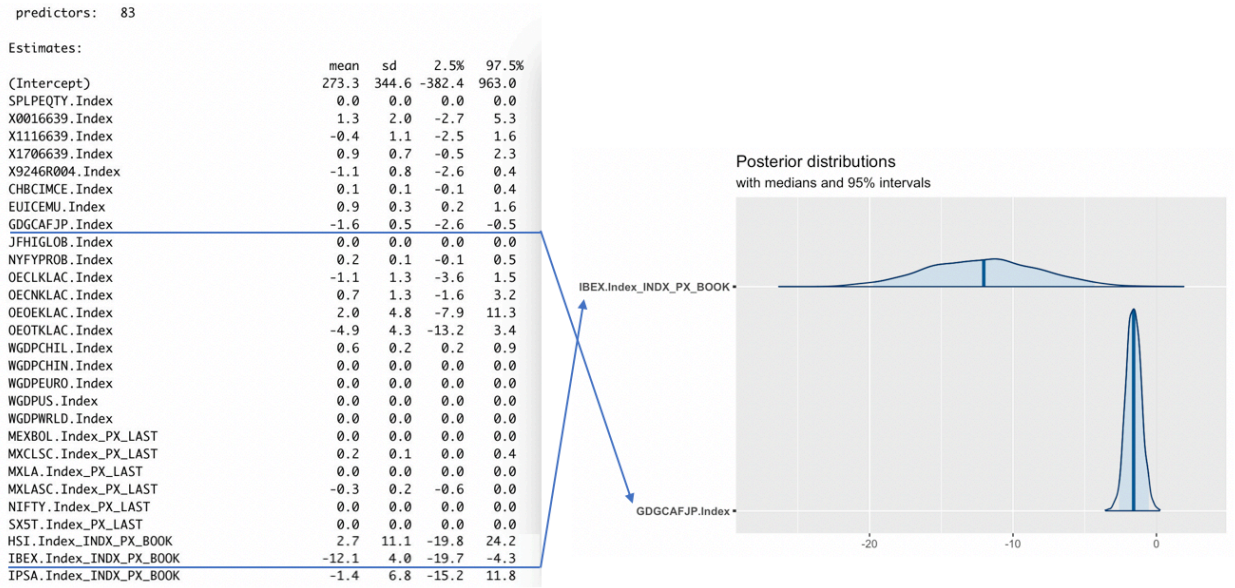
The graph above shows our model performance. The red line is the training loss and blue line is the test loss. The large disparity between these two lines is the sign of overfitting. With this result, our test RMSE is 0.15, which is pretty low but not accurate enough due to the issue of overfitting. One of the major reasons causing overfitting might be the quite small size of dataset we have since there's only 216 records in monthly data. In the future use, it would be better to include a lot more records in the model to solve the problem of overfitting and then getting a more accurate result. Perhaps, we can try daily dataset that has more than 6,000 observations.

6. Bayesian Model

Programming Language: R

The Bayesian model does not output the specific estimated coefficients, it gives out a distribution of estimated coefficients. It is a great application to infer the financial return. After such research, a package called BAS is chosen to build the model it provides easy to use functions to implement Bayesian model averaging in linear model. Based on the feature selection result, we modify the p-value and make it less than .00001 to ensure we gain the most pertinent indicators for each asset while preventing the model from crashing due to the limited computing power. As a result, each asset has roughly 80 to 90 variables. After important variable identification, we run the Bayesian prediction model for each asset with our selection of predictors.

Below is the summary and visualization of part of parameters in Private Equity model that includes about 83 predictors. We do not manually choose our prior but let the algorithms in the BAS package to choose for us.



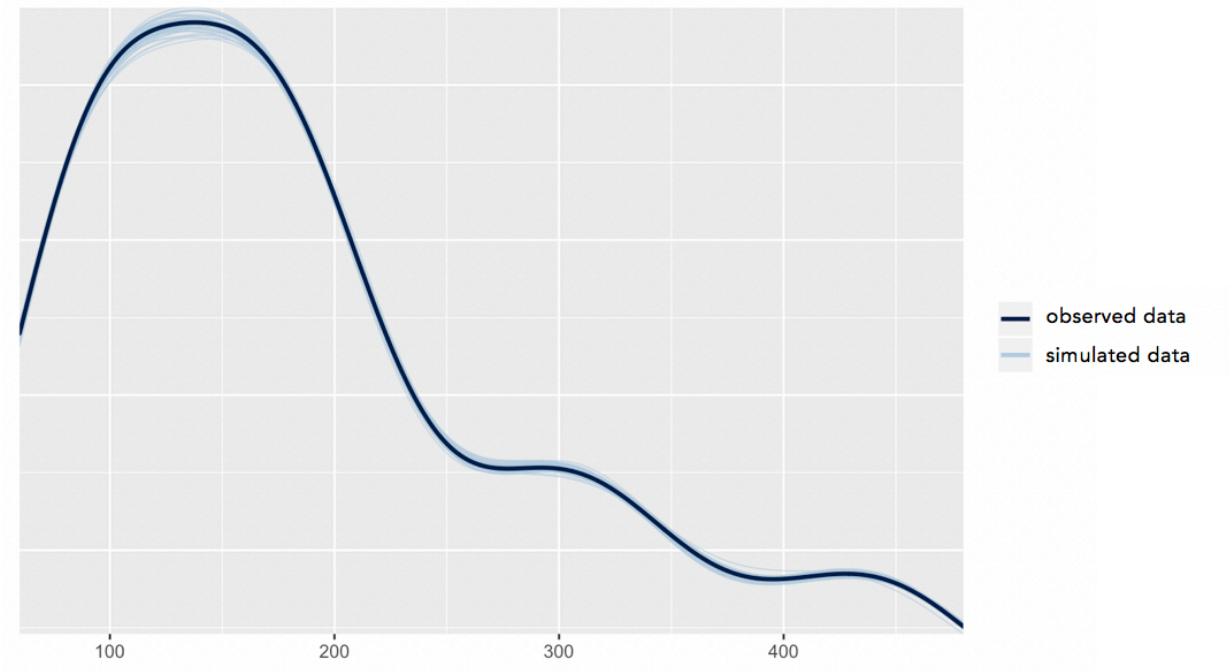
predictors: 83

Estimates:

	mean	sd	2.5%	97.5%
(Intercept)	273.3	344.6	-382.4	963.0
SPLPEQTY.Index	0.0	0.0	0.0	0.0
X0016639.Index	1.3	2.0	-2.7	5.3
X1116639.Index	-0.4	1.1	-2.5	1.6
X1706639.Index	0.9	0.7	-0.5	2.3
X9246R004.Index	-1.1	0.8	-2.6	0.4
CHBCIMCE.Index	0.1	0.1	-0.1	0.4
EUICEMU.Index	0.9	0.3	0.2	1.6
GDGCAFJP.Index	-1.6	0.5	-2.6	-0.5
JFHIGLOB.Index	0.0	0.0	0.0	0.0
NYFYPROB.Index	0.2	0.1	-0.1	0.5
OECLKLAC.Index	-1.1	1.3	-3.6	1.5
OECNKLAC.Index	0.7	1.3	-1.6	3.2
OEOEKLAC.Index	2.0	4.8	-7.9	11.3
OEOTKLAC.Index	-4.9	4.3	-13.2	3.4
WGDPCHIL.Index	0.6	0.2	0.2	0.9
WGDPCHIN.Index	0.0	0.0	0.0	0.0
WGDPEURO.Index	0.0	0.0	0.0	0.0
WGDPUS.Index	0.0	0.0	0.0	0.0
WGDPWRLD.Index	0.0	0.0	0.0	0.0
MEXBOL.Index_PX_LAST	0.0	0.0	0.0	0.0
MXCLSC.Index_PX_LAST	0.2	0.1	0.0	0.4
MXLA.Index_PX_LAST	0.0	0.0	0.0	0.0
MXLASC.Index_PX_LAST	-0.3	0.2	-0.6	0.0
NIFTY.Index_PX_LAST	0.0	0.0	0.0	0.0
SX5T.Index_PX_LAST	0.0	0.0	0.0	0.0
HSI.Index_INDX_PX_BOOK	2.7	11.1	-19.8	24.2
IBEX.Index_INDX_PX_BOOK	-12.1	4.0	-19.7	-4.3
IPSA.Index_INDX_PX_BOOK	-1.4	6.8	-15.2	11.8

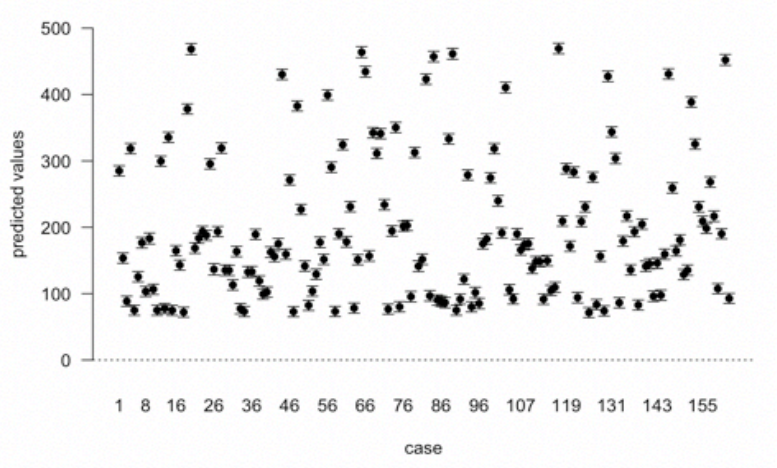
The graph above presents: the summary on the left side represents the 95% credible intervals, the mean and the standard deviation of the coefficient of each predictor. The summary can be translated to the coefficient distribution graph on the right. As we can see, the indicators with small range of the interval will result in a more narrow distribution while the larger range will generate a wider distribution. The wider the coefficient distribution is, the more uncertain how the predictor could impact the model performance.

After building the model, the first thing we did was to check the goodness of fit by simulating the data from the posterior predictive distribution which is resulted from the Bayesian model. Then, examine if the simulated data looks a lot like the data we observe.

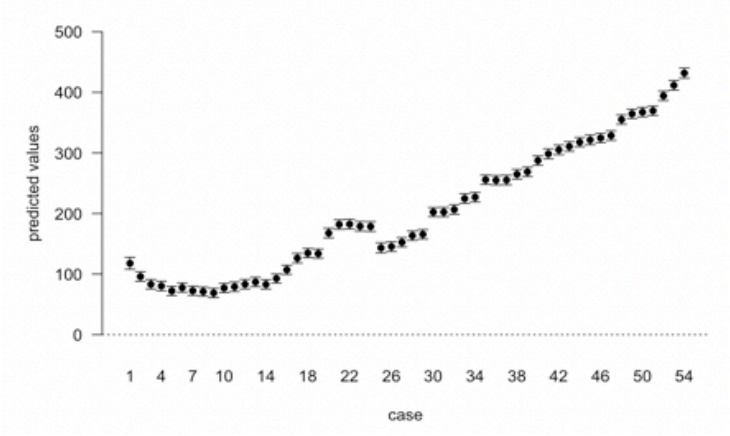


The graph above show the simulated data 50 times for Private Equity asset: the light blue on the plot is simulated data and the dark blue is the observed data. The visualization shows that the model fits really well.

When it comes to predicting, the Bayesian actually gives an estimate of distributions. However, we want to calculate the RMSE to compare with other models, so we take the mean of the parameter distribution serving as the estimate of the coefficient and generate the prediction. We get the train RMSE is 7.91 and the test RMSE is 7.90. The two plots below visualize the 95% credible interval of predicted values with an input data of train and test for Private Equity.



Credible Interval For Train
Train RMSE: **7.91**



Credible Interval For Test
Test RMSE: **7.90**

Bayesian Model Results

Model	Data Usage	Language	TEST RMSE				
			Private Equity	Real Estate	Buyout	Venture	Private Capital-Europe
Bayesian	Monthly	R	7.90	11.79	11.65	3.99	21.56
Number of Predictors			82	72	96	92	88

Above is a summary of the Bayesian model in all five chosen assets. Due to the limit of computing power, we are not be able to include all around 400 variables that we have into the Bayesian model, so we feature select for as many variables as we can to get almost the best RMSE for different assets here.

One of the important benefits of Bayesian is that we could include prior information to improve the accuracy. For the future, we can try changing the priors to some different numbers to see if it can give a better result. The only concern we have to keep in mind is the model tend to be intensive computationally when too many variables are input.

Key Takeaways

Model	Model Behavior	Data Usage	Language	TEST RMSE				
				Private Equity	Real Estate	Buyout	Venture	Private Capital-Europe
CAPM	Regression	Daily	R	**22.40**	**29.01**	**33.54**	**8.78**	**41.62**
BAYESIAN	Regression	Monthly	R	**7.90**	**11.79**	**11.65**	**3.99**	**21.56**
LSTM	Time Series	Quarterly	Python	**16.79**				
CNN	Time Series	Monthly	Python	**0.1537**				

Now that we see the different performance between 3 proposed models. Let's compare it to the benchmark model CAPM. Clearly, time series models LSTM and CNN produce much better results, however the main concern is overfitting that reduce their credibility in prediction. Our winner model here seems to be obvious, it is the Bayesian model.

Based on this summary, we withdraw 3 key takeaways:

- First: Say no to CAPM model, but re-consider its usage as a benchmark model instead
- Second: Apply Bayesian model to prediction that we know would yield better results with a confidence interval
- Third: while time-series model didn't work out this time. We believe there is a huge potential for its future success as a deep learning tool for prediction.

Future Implication:

With all of this in mind, we hope that AFP's Habitat will continue the effort of learning and testing model's outputs against actual results to identify the appropriate framework for each asset. Then from that experience, to tweak the model's features so that AFP Habitat can create a sophisticated network of analytics that will leverage the company's strategic decision to the next level.