# Introduction

This document is for organising and planning the design of the predictor. It will be added as an appendix for the thesis but not required reading.

# Requirement Spec

## Requirement Spec - Experiment Wrapper

1. Experiment wrapper must allow for different model types (NN, ANN, etc) to be cycled through.
2. Experiment wrapper must be able to run the following models:

* Convolutional NN
* Recurrent NN
* Generative Adversarial Networks
* Bagged Random Trees
* Neural Networks
* Artificial Neural Networks

1. Experiment wrapper must accept a dictionary of model parameters
2. Experiment wrapper must cycle through any model parameters passed as ranges and implement a combination with best scores
3. Experiment wrapper must train and test a model based on an input of (model/model parameters/input set)
4. Experiment wrapper must output the results of a single (model/model parameters/input set) as a dictionary.

## Requirement Spec - Financial Data Prep

Data Prep must add selected technical indicators

Data Prep, Sentimental Analysis - Twitter Only At the Moment

* + - 1. The data prepper must be able to accept an input informing it of the names of the following columns in the data master file:
      2. user id, tweet id, like count, retweet count
      3. The data prepper must be able to output all outputs as either a single value
      4. The data prepper must be able to output all outputs as either a single value for each individual user
      5. The data prepper should have the options to group people as cohorts and output their composite values together
      6. The data prepper must be able to accept a decay hyper parameter which reduces any sentiment value over time
      7. set of time series each tracking sentiment of a individual person
      8. The data prepper must be able to output a single "sentiment" value from -1 (negative) to +1 (positive)
      9. The data prepper must be able to filter users according to a overall frequency of posts
      10. The data prepper must be able to filter users according to a overall frequency of posts per year
      11. The data prepper must be able to optionally weight tweets according to likes
      12. The data prepper must be able to optionally weight tweets according to retweets
      13. The data prepper should be able to output a single value tracking mentions of various sentiments and concepts like:

fear, greed, excitement, risk, investment, volatility, liquidity, compliance, tweet volume, toxicity, volume of chatter, would increase the accuracy of stock predictors

filtered for tweets mentioning a given stock

The data prepper must output any data in a series of time increments, the range and

## Requirement Spec - Model

Model must accept multiple time series for training - completed PoC

Model should filter time series inputs to only files that contain a substring within their name any strings given from an input list - textbf{could be improved}

Model must allow the user to change between at least two models according to input from the experiment wrapper}

Model must run cross-validation on training set across the range of model parameters passed to it by the experiment wrapper and then train the final models on their best model parameterstextbf{ - could be improved}

Model must support k-fold cross-validation according to input parameters - textbf{could be improved}

Model must support bootstrap aggregation according to intput parameters - textbf{could be improved}

## Requirement Spec - Reporting

Reporting must inform of the % accuracy of an up-down prediction being correct for a variety of prediction lengths}

Reporting must inform on the % accuracy of an up-down prediction being correct for a variety of prediction lengths, only counting days where the confidence (or predicted increase) is above a user input value}

Reporting must inform on the estimated profit after using the predictions to lay simplified modelled long-short bets on the time history. Including inputs for:

fixed broker bet costs

variable broker bet costs

required confidence level to lay bet

confidence level required to maintain position

bet action lag

bet size to initial confidence coefficient

The reporting's simplified betting model should include functionality to increasing a bet in the case of increased confidence

Reporting must report the optimal model parameters and performance for each model type

## Requirement Spec - Design Notes

This section documents the design decisions that were taken in the model. In the current stage of development, these are up for contention and are in some cases just placeholder decisions that are deemed not too difficult to rectify later.

Design Decisions:

The CV finder study is run for each output to be predicted. This system needs review, from a computational an accuracy standpoint!

missing values are inferred to equal the previous value and are logged to ensure that not too many missing values are carried through

A check that no missing values in the index is needed

I have made the engineering judgment that the CV finder function in the state that I found it, instantiates the finder object for every run, deleting the research in the previous run

I believe the solution to this is to collect the run information from finder.cv\_results\_["param\_estimator\_\_alpha"] and other columns in this dictionary, to collect all the accuracy stats manually and then basing the best parameters on this info.

Each set of best model parameters (for regulatiation as such) found by the CV methods are grouped according to number of time steps, therefore in the current version of the model, every model predicting the same number of timesteps ahead, have the some reglusation parameters

I currently cut the dataset the length of the furthest prediction I make, this reduces the available data from the shorter timesteps, this doesn't seem like a major issue at the moment, needs to be reviewed later

# Modules

Below is an outline of the system, each following section will be named after the sections in the image

## Module - Design of Experiment

This section defines the stages of the design experiment, where parameters and their sweeps are defined. This is done as it will cause soft design lock-in for the available range of outputs available for the final results.

The model function will run

DoE definition

DoE wrapper

Model

Results

## Module - Experiment Handler}

## subDefine Design of Experiment}

subroutines: stock market data prep section

inputs: name of design experiment, model types and parameters dict, financial input data folder path, financial indicators

output: DoE orders dict, prepped financial input dict

This method gives the user an easy way of defining the DoE in a human-readable format and then outputting the DoE in a more complex data structure to allow the downstream programming to be more simple.

it will also prepare the training data, to save this from being repeated in the model section of the program.

THE EXACT DATA SCHEMA OF THE OUTPUT MUST BE DEFINED BEFORE PROGRAMMING

subRun Design of Experiment

input: DoE orders dict

subroutines: Run Model with Internal CV

output: results dict

subRun Model with Internal CV}

inputs: prepped financial input dict, model params

subroutines: prep model specific objects

outputs: predictions, figures, results reports

## Module - Stock Market Data Prep}

## subImport Financial Data}

inputs: target folder path list, index columns list, input cols to include list

outputs: df financial data

subpopulate technical indicators}

input: df financial data, stocks included list, technical indicators to add list

output:

subcreate step responses and split training test set}

input: df financial data, predictor output and ticker combos, pred steps list, test train split

output: X train, y train dict, X test, y test dict

subBlockingTimeSeriesSplit()}

Inputs: n splits

Outputs: <BlockingTimeSeriesSplit> btscv

This method creates the above class that is fed into XXX to define how the test set is time series is split up for training and testing.

## Module - Sentiment Analysis Data Prep}

subImport Sentiment Analysis Data Prep}

Inputs: target folder path list, index column name, filters dict, start datetime (optional), finish datetime (optional)

Outputs: df sentiment data

textit{This method needs to import, format and filer the required data. Before the rest of the sentiment analysis work.}

## Module - Model Training}

subreturn CV analysis scores}

inputs: X train, y train dict, CV Reps, btscv, cores used, params sweep dict

outputs: [CV history dict]

subreturn best model params from CV history}

# Data Schema

## financial\_input\_files\_dict}

# Design Considerations

# System Architecture

# Modelling Questions

end{document}

----- End of Document -----

# Config Control

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 0.0.1 | 10/05/2023 | Initial population from overleaf assets |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |