Capstone Proposal

Proposal for solving a problem by applying **machine learning**.

Capstone Proposal ***1***

Proposal for solving a problem by applying machine learning. *1*

1.0.- Domain background, Introduction *1*

2.0.- Problem statement *2*

3.0.- Datasets and Inputs *2*

4.0.- Solution Statement *3*

5.0.- Benchmark Model *3*

6.0.- Evaluation Metrics *4*

7.0.- Project Design *4*

7.1.- Dataset Selection: *5*

7.2.- Exploratory data analysis: *5*

7.3.- Feature Engineering: *5*

7.4.- CV Strategy: *5*

7.5.- Model Selection / Training: *5*

7.6.- Model Hyperparameter Optimization: *5*

7.7.- Model Validation: *5*

7.8.- Model Deployment: *5*

8.0.- Final Remarks *6*

1.0.- Domain background, Introduction

Stock market analysis/prediction is one of the most challenging fields to estimate due to the multi-dimensionality and complexity of the inputs. There can be many factors involved in the calculations varying from multiple company/competitor interactions and variability, public perception, rational and irrational behaviour of traders and many more, the combination of all these aspects together make price volatile and challenging to predict.

Data scientists and Financial theorists for over half a century have been employed to make sense of the stock market and its variations in order to increase return on investments, but it has been an overwhelmingly difficult challenge for humans to solve due too the complexity and massive amount of inputs

*Thanks to the advancements in **machine learning** algorithms and its applications, the field has evolved to combine multiple inputs like never before. Current state of the art approaches join information from the organization historical price trends, technical data, social media, news and classic techniques utilizing non-deterministic solutions that can "learn" what is going on from millions of data inputs, these techniques can*

utilize a mix of neuronal network models that rely on long short term memory units and natural language understanding. unearthing patterns and insights we didn't see before, using them to make unerringly accurate predictions.

I believe by building a system that can accurately prognosticate stock market variations we can use it to successfully predict a stock's future price maximizing investor's gains.

2.0.- Problem statement

*Due to the multi-dimensionality and complexity of the problem the main objective of this capstone project is to provide a **machine learning** model that utilizes state of the art algorithms, historical stock market data of a public trade company to predict future trends.*

We will start simple utilizing one single attribute and expand to increased complexity by incorporating more inputs, moving from simple algorithms like linear regression to advanced techniques like auto ARIMA and LSTM Models.

To make machine learning predictions quantifiable, measurable and replicable, we will define the datasets utilized and provide access to them, select a well-known performance metric to quantify improvements over a baseline model and provide the codebase (hosted on source control platform) to replicate the results.

3.0.- Datasets and Inputs

We will be using statistical figures to identify trends on the market, the datasets utilized will be obtained from Quandl; Quandl (https://www.quandl.com/) is a platform source for financial, economic, and alternative datasets, serving investment professionals

Yahoo! Finance (https://ca.finance.yahoo.com/), is a platform that provides It provides financial news, data and commentary including stock quotes, press releases, financial reports, and original content.

The data that we will use is from J.P. Morgan Chase & Co. we are planning to utilize a total of 10 years of data were 7 years will be used for training and 3 years for validation and testing

the data can be obtained with the following link. https://ca.finance.yahoo.com/quote/JPM?p=JPM

In order to predict the movements of J.P. Morgan Chase & Co. We will use daily data to build the model and we will separate our datasets in a train, validation and testing allocating 70%, 20% and 10% of the data respectively, we will utilize years of information. The main dataset will be enhanced with the utilization of correlated assets and technical indicators.

The J.P. Morgan Chase & Co. dataset follows this structure date, open, high, low, close, adj close and volume

Currency in USD

Date	Open	High	Low	Close*	Adj Close**	Volume
Jul. 19, 2019	114.89	115.12	113.40	113.54	113.54	10,401,400
Jul. 18, 2019	113.93	115.07	113.55	114.67	114.67	9,400,700
Jul. 17, 2019	114.43	114.94	113.73	113.99	113.99	13,120,900
Jul. 16, 2019	113.48	115.50	112.92	115.12	115.12	16,945,000
Jul. 15, 2019	115.54	115.57	113.53	113.90	113.90	12,946,600
Jul. 12, 2019	114.13	115.35	113.93	115.30	115.30	10,783,400

*The columns **Open** and **Close** represent the starting and final price at which the stock is traded on a particular day.*
***High**, **Low** represent the maximum and minimum price of the share for the day.*
***Volume** is the number of shares bought or sold in the day and Turnover (Lacs) is the turnover of the particular company on a given date.*

We will consider closing price as the target variable

4.0.- Solution Statement

In order to predict the movements of J.P. Morgan Chase & Co. We will develop multiple models that increase in the level of sophistication this will ensure that our development strategy is progressive and we make significant steps toward a final solution. To measure consecutive improvements in the development in a quantifiable way we will utilize the same performance metric allowing this project at the same time to be replicable

The solutions will range from basic linear model to the utilization of Long Short Term Memory networks (LSTM)

The objective of the machine learning model will be to predict the performance of the stock price within a certain time in the future, the selected target for the model will be the stock close price at time t and we will predict the price of the stock at time t+1 using previous days of historical stock market information t-n and multiple other features that will be developed during the exploratory data analysis phase.

*The solution will be implemented in **TensorFlow** or **PyTorch** and deployed with **AWS SageMaker** utilizing a combination of the AWS solutions to reach an end to end estate, the project will consider or utilize:*

1. *Amazon SageMaker / Jupyter lab*
2. *Lambda Functions*
3. *S3 Storage*
4. *API Gateway*

5.0.- Benchmark Model

Establishing a baseline is essential on any time series forecasting problem; to establish a performance baseline on our stock market prediction problem, we will focus on the following steps to ensure replicability with defined and measurable outcomes:

1. *The dataset that we are going to use to train and evaluate models needs to be easy to access.*
2. *Have a resampling technique to use to estimate the performance of the model (e.g. train/test split).*
3. *Have a defined performance measure to use to evaluate the predictions (e.g. mean squared error).*

To create a baseline/benchmark model with our time series dataset we will utilize the persistence algorithm. There are specific requirements that we need to be aware of this requirement of a good technique for making a baseline forecast are:

1. ***Simple**: A method that requires little or no training or intelligence.*
2. ***Fast**: A method that is fast to implement and computationally trivial to make a prediction.*
3. ***Repeatable**: A method that is deterministic, meaning that it produces an expected output given the same input.*

The persistence algorithm uses the value at the previous time step (t-1) to predict the expected outcome at the next time step (t+1).

6.0.- Evaluation Metrics

After our machine learning model has been trained is quite important to assess how well the model it is able to capture patterns and predict, in order to diagnostic our model we will utilize evaluation metrics and residual diagnostics

For our model we will utilize:

1. ***RMSE**: Root mean squared error, this metric is typically used in regression problems and works quite well as an indicator of performance*

$$RMSE = \sqrt{MSE}$$

2. ***MAPE**: Mean absolute percentage error, it is scale-independent and represents the ratio of error to actual values as a percent*

$$MAPE = mean\left|e_t / y_t\right|$$

While evaluation metrics help determine how close the fitted values are to the actual ones, they do not evaluate whether the model properly fits the time series. Instead, the residuals are a good way to evaluate this

1. *The residuals are uncorrelated (Acf = 0)*
2. *The residuals follow a normal distribution, with zero mean (unbiased) and constant variance* $(e_t \approx N(0, \sigma^2))$

7.0.- Project Design

*Our final machine learning model will be built utilizing neuronal networks, the define units will be Long Short Term Memory Cells (LSTM) created with one of the two most popular neural network libraries (**TensorFlow** or **PyTorch**), the architecture of the final model will be defined upon multiple iterations and tests in a loop with hyperparameters optimizations in order to improve model performance.*

*To make the model quite effective we decided to select as our final architecture the utilization of **LSTM** because they widely use for sequence prediction problems and proven to be extremely effective in this field. The reason they work so well is that **LSTM** is able to store past information that is important and forget the information that is not. **LSTM** has three gates:*

1. ***The input gate:*** *The input gate adds information to the cell state*
2. ***The forget gate:*** *It removes the information that is no longer required by the model*
3. ***The output gate:*** *Output Gate at LSTM selects the information to be shown as output*

The project will consider the following steps in order to solve the problem:

7.1.- Dataset Selection:

The information will the store as a form of CSVs in the Cloud, we will utilize a define S3 storage location for the files to make the replicability of the work easy

7.2.- Exploratory data analysis:

The information will be analyzed in detail in order to remove or eliminate outlier, at the same time we will explore the information to build a solid understanding of the characteristics of the data, for this stage we will use Amazon SageMaker

7.3.- Feature Engineering:

In this stage of the model based on the analysis completed in the EDA stage, we will build new features in order to increase the model performance, we will test how each of these features impacts the model utilizing the benchmark models as a baseline, for this stage we will use Amazon SageMaker

7.4.- CV Strategy:

In this stage we will separate the datasets based on the previous section descriptions, were we defined to utilize a 70% /20% /10% rule of distribution within Train, Validation and Testing strategies this will ensure the model evaluation stay consistent and provide realistic outcomes, for this stage we will use Amazon SageMaker

7.5.- Model Selection / Training:

During this stage we will train a simple architecture starting with a basic number of cells and layers in the model, we will complete the training and evaluation to see how it performs, for this stage we will use Amazon SageMaker

7.6.- Model Hyperparameter Optimization:

In this stage of the training, we will deploy an optimizer to test multiple architecture and configurations to find the best combination of parameters and architecture for our model, for this stage we will use Amazon SageMaker

7.7.- Model Validation:

During this stage we will utilize the test datasets to validate the final model after training and hyperparameter tuning has been completed, this will provide a reliable measure of the performance of the model in never seen data before, for this stage we will use Amazon SageMaker

7.8.- Model Deployment:

The final model after all the test has been completed will be deployed utilizing amazon SageMaker and Gateway, for this stage, we will use Amazon SageMaker

8.0.- Final Remarks

*We believe that by following all the previous steps in an organized and structured way, we can achieve outstanding results from the point of view of replicability, creating a model that will be quantifiable and measurable. We have decided to utilize the Data Science Cookie Cutter standards to construct this model (https://drivendata.github.io/cookiecutter-data-science/)and as for development workflow we will focus on all the **AWS** suite of tools for machine learning models, majority of the development will be completed on Amazon **SageMaker** and the code will be hosted on **GitHub**.*