Geometric Progression

We Bring AI to Reality.

The AI Accelerator Programme

Overview

Our AI Accelerator Programme is designed to guide you through the journey of mastering AI, regardless of your starting point. This comprehensive course collection is your ticket to becoming proficient in programming and modelling, and ultimately, a leader in the AI field.

The course is divided into a series of modules such that delegates with previous experience are only required to attend the modules which are of interest to them.

The core modules are divided into:

1. **Python programming essentials**
2. **Core Python libraries.**
3. **The "Quants" for AI and Quant Portfolio metrics with Python.**
4. **An Overview of AI, Machine Learning, Deep Learning & Core AI Jargon.**
5. **The Art of Modelling using Python for AI.**
6. **Machine Learning with a Financial Markets Focus**. This section includes the traditional Machine Learning Algos including ensembles and XAI.
7. **Deep Learning with a Financial Markets focus.** It is important to point out that the deep learning section of the course can be run using Tensorflow or Pytorch or both.
8. **Unsupervised Learning with a Financial Markets focus.**

Other optional modules include:

- **Time series analysis (recommended**). This section includes traditional methods and AI methods.
- **Reinforcement Learning.**
- **Anomaly detection and Regime Change** (including Markov chains).
- **Algo Trading & Portfolio Management.**
- **Automation & Webscraping etc**.
- **Implementing LLM's**.

Each of the above modules can be run separately, independently and customized to your needs.

Hands-On Learning Experience

Our programme emphasizes learning by doing. You'll engage in live online lectures that can be customised by yourselves. Expect fun projects and occasional homework to reinforce your learning.

Course Requirements

Prerequisites:

- While a background in elementary mathematics & statistics at the university level is beneficial, it's not mandatory. We provide a comprehensive overview of the required mathematics and statistics if required.

Delegates who attend this course will receive all notes electronically. This includes the slides, videos, the code templates with solutions and many links to excellent further reading. All "lectures" are recorded and are hosted exclusively online using Teams.

Tutor: Mark Raffaelli

Mark is obtained his CFA Charter in 2000 and became fellow member of the Global Association of Risk Professionals (FRM) in 2001. Mark's extensive experience includes:

- Trading in Spot & Derivative Products professionally.
- Development of quantitative financial models for Surveillance, Performance Attribution, Price Validation, Price Models, Risk and Automation.
- Developments of Apps for the investment and insurance industry.
- Machine Learning and Deep Learning with Sklearn, Tensorflow & Pytorch in the Financial Markets for Banks, Asset Managers and general business.
- Time Series Analysis, Regime Change & Optimization with and without AI.
- Strategic implementation of AI.

Those who have attended Mark's courses will know about his passion and ability to cut through jargon, simplify technical issues and provide real life examples.

Part 1: Learning to Code in Python with a Focus on AI and the Financial Markets.

Delegates will learn by coding themselves. We make use of extensive examples that relate to the financial markets, modelling, econometrics and factor models.

Once delegates have grasped the core fundamentals of coding independently, we will accelerate the learning process by using AI to assist us with coding.

You can customise the agenda to focus on alternative areas if required.

Detailed Agenda & learning outcome statements:

1. The stuff you need to know before we start.

Delegates will be able to:

- Understand why it is so important to know how to Program in today's world.
- Explain why Python is the language of choice for machine learning and deep learning.
- ***Distinguish between Modelling versus professional programming.***
- Navigate and efficiently use Jupyter notebooks.
- Create and use Colab notebooks.
- Create comments in notebook cells.
- Create markup cells to create a rich user experience incorporating markup language for headings, paragraphs, images, links etc.

2. Datatypes

Delegates will be able to create, use and explain of the different types of datatypes:

- Use the print statement and the type statement.
- Create a string and output string.
- Output a string using a host of different string formatting methods.
- Create, use and explain integers.
- Create, use and explain floating type numbers.
- Create code to cast from one datatype to another.
- Create and use variables.

3. The World of Functions

Delegates will be able to:

- Create a function.
- Understand and correctly indent code to make the function work correctly.
- Understand and use white-space appropriately.
- Define the function Signature.
- Create arguments and parameters and understand the difference.
- Create and use “docstrings”.
- Create user-defined functions.
- Create, use and understand default arguments and optional arguments.
- Create and implement python built in functions.

4. Scope

Delegates will be able to:

- Understand and define the difference between local and global scope.
- Understand the concept of enclosing scope and variable clash.

5. Conditional logic

Delegates will be able to:

- Create an if statement and use the else-if and else Logic.
- Put together some nested if statements.
- Understand the concept of XOR

6. Loops

Delegates will be able to:

- Define, create and use the range function.
- Create a for loop and understand the logic of how the code is processed.
- Create a while loop and understand the concept of incrementation.
- Create nested loops

7.Useful Loop Functions

Delegates will be able to:

- Solve algorithmic problems using flags.
- Implement the “break” out of a loop code.
- Implement the “continue” loop code.
- Use and understand the “pass” place-holder.
- Understand the concept of the “iter” method and implement it.

8. Lists

Delegates will be able to:

- Use and create lists.
- Slice a list.
- Append, delete and extend lists.
- Pop an item off a list.
- Loop through items of a list.
- Implement list functions.

9. Sets & Tuples

Delegates will be able to:

- Create a set and implement set operations.
- Use a set to derive unique items.
- Use set mathematics to filter data.
- Create a tuple.
- Implement tuple functions.

10. Dictionaries

Delegates will be able to:

- Create, use and develop a dictionary.
- Use economic stats to fill a dictionary.
- Slice a dictionary.
- Implement a dictionary and list comprehension.
- Loop through the dictionary.

11. Modules

Delegates will be able to:

- Understand and perform a pip install.
- Import external modules.
- Import self-created modules.
- Create a ”namespace”.
- Import modules with wild cards.
- Create a module.
- Import “Pixiedust” and use it for debugging.

12.Visual Studio Code

Delegates will be able to:

- Independently set up the Visual Studio Code environment.
- Download a python extension.
- Select a python interpreter.
- Implement debugging such as step through, step over and break points

13. Automation

Delegates will be able to:

- Use the pyautogui module.
- Automate programs with mouse clicks, event handling and keyboard handling.
- Use automation to scrape values from the web.
- Run the “mouse now” program to get screen coordinates.

14. Functional Programming and Lambda Functions

Delegates will be able to:

- Understand and use a first-class function.
- Pass a function as an argument.
- Create a lambda function.
- Implement the “filter” statement on a lambda function.
- Implement the “map” statement on a lambda function.
- Implement a “reduce” statement on a lambda function.

15. Object Orientated Programming (OOP)

Delegates will be able to:

- Explain the difference between procedural code and object orientated code.
- Create a class.
- Encapsulate functions inside a class.
- Create an object constructor.
- Create an object instance.
- Call an instance attributes and methods.
- Implement dunder methods.
- Using “self” in code.
- Build their own OOP Calculator.

16. Wrapping up with Environments, Zipping, widgets and decorators.

Delegates will be able to:

- Set up a Python Environment.
- Use Jupyter widgets
- Use the Python Zip function
- Understand and use @Decorators.
- Create an “.exe” using pyinstaller.

Part 2 Core Python libraries

1. Integrating Python with Excel & connecting to SQL.

Delegates will be able to:

- Perform a CRUD with Openpyxl.
- Use the basics of Xlsxwriter
- Run Python from Excel with XLwings
- Running Python inside native Excel. This only currently works for those delegates part of the Microsoft 365 Insider Program – soon to be released globally.
- Connect to an SQL database and write CRUD queries.

2. Working with Dates and times.

Delegates will be able to:

- Implement and use the Datetime module.
- Implement and use the DateUtil module.

3. Numpy.

Delegates will be able to:

- Understand why Numpy is so important.
- Convert different datatypes to numpy arrays.
- Measure Array shapes, dimensions and size.
- Use vectors in numpy.
- Portray images as vectors.
- Reshape, ravel and flatten arrays.
- Implement Numpy functions and seeding.

4. Pandas Part 1

Delegates will be able to:

- Convert datatypes to a Pandas series.
- Create, update and delete a series.
- Implement Series functions and filter a series.
- Convert datatypes to a pandas dataframe.
- Create, update and delete a dataframe.
- Implement dataframe functions.
- Distinguish between the commands “display vs print”.
- Implement different reference conventions (loc and iloc etc).
- Perform selection, filtering and assignment with loc, iloc, at,iat.

5. Intermediate Pandas.

Delegates will be able to:

- Loop through big data with Pandas.
- Use superior alternatives to loops with referencing with Pandas.
- Import data from, csv, Excel and pdf into Pandas.
- Perform HTML Webscraping with Pandas.
- Create code for remote access of data with Pandas.
- Implement Pandas TimeSeries and datetime.
- Build an algo trader and backtest the results.
- Perform data interpolation with pandas.
- Output to Xlsxwriter with pandas.

6. Plotting Charts with Python.

Delegates will be able to:

- Chart with Matplotlib.
- Plot points, scatterplots and a line.
- Add some labels and features to the graph with code.
- Plot a barchart with code.
- Plot a pie chart with code.
- Plot a stacked chart with code.
- Create multiple plots with subplots.
- Create variations of subplots.
- Add axes manually in code.
- Use formulas to plot with Numpy.
- Create 3D Plots.
- Rotate the plots and Azimuth's.
- Create quick plots from pandas to matplotlib.
- Charting with Seaborn and Plotly.

7. Creating a Web app and GUI with Python

Delegates will be able to:

- Create a webapp using Streamlit.
- Use PysimpleGUI and other modules to create a graphical user interface.

Part 3: The "Quants" for AI and Quant Portfolio metrics with Python.

The purpose of this section is to go through the calculus, statistics, linear algebra and optimization that you would need to understand machine learning and deep learning algorithms. We also take a look at the useful portfolio modules that do portfolio optimization and calculate many of the portfolio metrics such as Sharpe, M^2, Sortino, upside and downside risk, drawdown etc.
This module can be structured in 3 separate ways where we either:

- Provide a detailed recap of the underlying sections of mathematics and statistics theory, followed by detailed examples using the respective Python modules. This is meant for those delegates that have not previously done mathematics and statistics following university or school and have pretty much forgotten the detail.
- Provide a brief recap of the underlying sections of mathematics and statistics followed by detailed examples using the respective Python modules. This is meant for those delegates who are competent at mathematics and statistics and the concepts are familiar.
- Assume all delegates are familiar with the underlying mathematics and statistics and simply demonstrate how to use the respective Python modules to do the calculations.

In the learning outcome statements below, I have erred on the side of brevity, but it is important to note that we dive to a significant amount of detail and explain many of the concepts in plain English. In all cases we implement examples and solutions using Python.

Agenda & Learning Outcome Statements.

1. Calculus for AI.

Delegates will be able to:

- Recap some of the math you did at school but can't remember.
- Define functions and function properties including: inverse, continuous, piecewise, monotonic, etc.
- Describe composite functions.
- Understand and do calculations with exponents radicals and surds with Python.
- Understand and do calculations with Logs, Ln and e.
- Implement NACA,NACQ,NACM,NACD to Continuous compounding
- Use Numpy in Python to implement of the above math.
- Understand the big deal of "lnx" and "-lnx" in machine learning and the financial markets.
- Implement Partial derivatives with Python.
- Use Sympy to solve equations.
- Use Sympy to do differentiation and integration.
- Understand LATEX notation and implement it.
- Display math notation in Python notebooks.

2. Linear Algebra for AI.

Delegates will be able to:

- Understand vectors, matrices and scalers in the context of tensors.
- Define Linear combinations and linear independence.
- Implement Vectors in Python.
- Define the Dot Product and implement in Machine learning.
- Apply the +1 trick to Regressions.
- Define Tensor dimensions.
- Create 1 d array vs 2d (matrices) and more dimensions.
- Calculate the Matrix inverse in numpy and Excel.
- EigenValues and Eigenvectors in numpy.
- Generalised Additive Models and basis splines.
- Interpolate yield curves with linear, quadratic and cubic splining using code.
- Take the first and second derivatives for smooth transitions in curves.

Statistics for AI.

Delegates will be able to:

- Implement Scatterplots, Relplots, Boxplots, IQR and Quantiles using matplotlib.
- Change colour maps and styles in matplotlib.
- Calculate the Arithmetic mean, geometric mean, harmonic mean and their properties.
- Plot alternative distributions with Swarm Plots, Violin Plots and Strip Plots, Histograms and Displots.
- Recap the concepts of probability.
- Use probability for arbitrage.
- Recite the probability rules.
- Understand the different types of distributions and their characteristics.
- Understand Kernel density estimation.
- Implement Pairplots with Seaborn
- Choose palettes and cmaps with Seaborn.
- Work with Facet grids.
- Implement the QQplot with statsmodels.
- Use Seaborn to generate Heatmaps.

4. An Introduction to Optimisation.

Delegates will be able to:

- Objective functions and constraints.
- Convergence
- Local versus global optimisation
- A look at some elementary optimisation algorithms.

5. Portfolio Manager modules

Delegates will be able to:

- Sources financial data modules.
- Calculate the risk of a portfolio.
- Calculate Covariance, Correlations in the context of portfolio optimization.
- Implement Covariance and correlation matrices and implementation with Seaborn.
- Run a host of Quant metrics such as Sharpe, M^2, drawdown, Sortino, Omega etc. with a host of libraries.
- Algo trading and Algo trading libraries.
- Backtesting and backtesting models.
- Valuation libraries of different asset classes including derivatives.
- Portfolio optimization using mean-variance, Black-Litterman, Expected Shortfall Optimization etc. Note other optimization models are introduced in the relevant AI models including factor based optimisation, Hierarchical Risk Parity etc.
- Yield curves and yield curve interpolation.
- Bootstrapping.
- Interest rate modelling.
- Pro Forma Financial Modeling

Please note that this Section / module can be customised based on the requirements and knowledge of the delegates.

Part 4: An Overview of AI, Machine Learning, Deep Learning & Core AI Jargon.

This module is tailored for delegates who are eager to learn how to implement AI through machine learning and deep learning. It offers a comprehensive overview of how these concepts interconnect, covering essential terms and key ideas you need to grasp before delving into the detailed implementation of the algorithms. Delegates will be introduced to various algorithms at a high level, without the need for coding. The presenter will also demonstrate the concepts being taught, highlighting the benefits of applying these algorithms in future projects.

1. The AI Revolution in Business:

- Definition of AI & a brief history of artificial intelligence.
- Key developments and triggers of AI's recent surge.
- Impact of AI on various industries.

2. Demystifying AI Jargon and Learning Models

Explanation of key AI terminologies such as:

- ANI, AGI, ASI, Machine learning versus Deep Learning.
- Neural networks, LLM's, GAN's, Transformers, Reinforcement Learning.
- Understanding Chatbots such as ChatGPT, Bard etc.
- Microsoft Copilot And much more!
- Real-world examples illustrating each concept.

3. AI in Business Strategy

- How AI is reshaping business models.
- The role of AI in enhancing customer experiences.
- AI's influence on competitive landscapes.

4. Practical Machine Learning Algorithms, Neural Networks and Advanced AI Technologies

- "Black Box" models versus "White Box" models.
- Live interactive demos of Machine Learning algorithms and numerous Neural Networks, Chatbots and Copilot etc.
- Case studies on business applications of these algorithms
- Discussion on selecting the right algorithm for specific business problems.
- Overview of the latest advancements in AI technologies.
- Interactive session on the implications of these technologies in business.

5. Risks, Ethics and Implications of AI:

- Discussion on ethical considerations in AI.
- Identifying and managing risks associated with AI in business from security to staff re-structuring.
- Case studies on ethical dilemmas and risk mitigation What are people saying about AI?

6. Managing AI Transformation.

- Frameworks for implementing AI in your organization.
- Steps for managing the transformation process.
- Best practices for ensuring sustainable AI integration.

7. Interactive Q&A and Wrap-Up.

- Open floor for questions and personalized insights.
- Creating an AI-centric culture within your organization.
- Discussion on specific business challenges and opportunities in AI.
- Summary of key takeaways and next steps.

Part 5: The Art of Modelling for AI

This is a short but important section that deals with the fundamentals of building models for AI.

1. Introduction to building Algorithms and models.

Delegates will be able to:

- Building a model: Art or Science
- Knowing your data.
- Model per-requisites.

2. Data Pre-processing and implementing models

Delegates will be able to:

- Understand the types of data and casting in Pandas.
- Use the Pandas data cleaning tools.
- Deal with NAN's gracefully.
- Interpolate missing data.
- Provide an overview of the Sklearn Library.
- Interpolate with Sklearn SimpleImputer.
- Understand the big picture of machine learning.
- Use the KNN imputer tool in SKLEARN.
- Implement Label encoding with Sklearn.
- Use One hot encoding with Panda' s and Sklearn.
- Define and prevent the dummy variable trap.
- Implement transforms and inverse transforms.
- Use column transformer in Sklearn.
- Differentiate between Supervised vs Unsupervised learning.
- Implement SKLearn Train Test Split
- Understand orders of magnitude.
- Implement feature Scaling and use the different feature scaling methods.
- Use Sklearn for MinMax scaling.
- Understanding and implementing Cross validation.

Part 4: Machine Learning with a Financial Markets Focus

Note, delegates will learn by coding themselves. We make use of extensive examples that relate to the financial markets, modelling, econometrics and factor models. You can customise the agenda to focus on alternative areas if required.

1. Linear Regression

Delegates will be able to:

- Elementary linear regression.
- Define the quadratic cost.
- Define and implement the SSE / RSS.
- Create a loss function in Excel.
- Implement Linear regression in Sklearn.
- Implement Linear regression in Statsmodels.
- Work out R squared, coefficients and intercept.
- Make predictions from the regression model.
- Understand the difference between Endogenous versus Exegenous.
- Use Scipy to do linear regression.
- Explain alpha, beta and factor models.
- Derive Alpha and beta from security returns with Sklearn.
- Implement the famous ISLR media dataset.
- Calculate the MAE vs MSE VS MSLR.
- Calculate the Adjusted R squared.
- Understand overfitting versus underfitting with a financial markets example.
- Understand Bias versus Variance.
- Explanation of the Bias Variance tradeoff.

2. Multiple regression and Regression assumptions

Delegates will be able to:

- Define Degrees of freedom
- Implement Correlation versus R Squared in sklearn.
- Implement a Multiple regression model in sklearn
- Implement a Multiple regression model in statsmodels.
- Output charts in 3D for multiple regression models.
- Create Randomness and correlation plots.
- Use python to determine Multicollinearity
- Use python to find Homoscedasticity vs Heteroskedasticity.
- Use python to determine mean of residuals measure.
- Use python to determine autocorrelation / serial correlation.
- Use python to determine independent variables that are uncorrelated with the error term.

3. Stepwise Regression

Delegates will be able to:

- Create a Normal distribution of error terms.
- Unpack feature selection.
- Use hypothesis testing, p values, alpha.
- Implement stepwise regression with Statsmodels.

4. Regularization

Delegates will be able to:

- Recap the L1 and L2 norms.
- Define and use Lasso Regression with sklearn and keras.
- Define and use Ridge Regression with sklearn and keras.
- Define and use Elastic Net Regression with sklearn and keras.
- Implement regularization in ML models and neural networks.

5. Polynomial Regression

Delegates will be able to:

- Understand Polynomials
- Implement Polynomial regression with Sklearn
- Use Numpy polyfit.
- Determine the most appropriate degree.
- Use the coronavirus dataset to predict the next wave of covid.

6. Logistic Regression

Delegates will be able to:

- Use Logistic Regression in sklearn.
- Distinguish between Binary versus Multiclass classification.
- Define the Odds Ratio.
- Derive the sigmoid function.
- Derive the Logistic Regression Loss function.
- Make Prediction using the Toy Datasets.

7. Evaluation of Classifiers

Delegates will be able to:

- Understand grayscale and RGB in numbers.
- Understand Accuracy in Machine learning.
- Create Baseline Methods in Sklearn.
- Implement the confusion matrix and define True positives, False Positives, False Negatives, True Negatives.
- Define and calculate Type I and Type II errors.
- Implement confusion matrices in Python for binary and multiclass.
- Understand Receiver Operating Curves and implementing them in Python.
- Understand and implement Precision and Recall curves in Python.
- Calculate the F1 score, F betas scores and understand the relationship with the harmonic mean.

8. K nearest Neighbours

Delegates will be able to:

- Understand and implement K Nearest Neighbours (KNN) in Python.
- Calculate the Hamming Distance.
- Calculate the Minkowski Distance.
- Calculate the Vector Norm.
- Plot a decision boundary with Matplotlib and MLextend.
- Determine the appropriate number for K.

9. Naïve Bayes

Delegates will be able to:

- Understand and implement the Bayes Formula.
- Calculate Conditional Probability.
- Define and understand the Monty Hall problem.
- Distinguish between dependent versus independent events
- Understand why the algorithm is called Naïve Bayes and the formula.
- Implement Naïve Bayes in Python

10. Support Vector Machines

Delegates will be able to:

- Calculate and define the Maximal Margin Classifier
- Understand the softening the margin in MMC.
- Implement a Support Vector Classifier in python.
- Understand and use the SVM Error Function.
- Define and use the C and Gamma hyperparameter.
- Implement the Polynomial Kernel.
- Implement the Radial Basis function Kernel
- Implement SVM in Python
- Implement the different SVM methods on toy datasets with python.
- Plot the decision boundary with varying values for C and Gamma.
- Use RBF kernel to identify people in images.
- Implementing SHAP on SVM.

11. Decision trees

Delegates will be able to:

- Understand greedy algorithms.
- Create a Recommendation system with decision trees.
- Understand and use the Gini Impurity.
- Understand entropy and implement it in decision trees.
- Understand when do you stop building a tree.
- Plot a tree in python sklearn.
- Plot a tree with graphviz.
- Plot a tree with dtreeviz.
- Implement confusion matrices and decision boundaries with decision trees
- Implementing SHAP and XAI on decision trees.

12. Ensemble Learning

Delegates will be able to:

- Understand the concepts of ensemble learning.
- Define and implement Base Learners and meta learners.
- Implement Bagging with sklearn.
- Implement Boosting with sklearn.
- Implement Stacking with sklearn.
- Implement a Random forest with sklearn.
- Implement AdaBoost.
- Implement Gradient Boosting.
- Implement XGBoost.
- Implement Ensemble learning with Stacking.
- Using the different ensemble methods to plot a decision boundary.
- Create stacking variations practice in sklearn using different algorithms.
- Implementing SHAP and XAI on ensembes.

13. Implementing K Folds Cross Validation and Hyper Parameter Optimization.

Delegates will be able to:

- Recap the concept of degrees of freedom.
- Recap the weaknesses of train, test, split.
- Understand and describe Cross Validation.
- Implement K folds cross validation in sklearn and a variation thereof in Keras.
- Implement a hyperparameter optimization with gridsearch with sklearn.
- Pickle machine learning files.
- Implement hyperparameter optimization in Keras.
- Implement hyperparameter optimization with a pipeline using multiple algorithms in skLearn.
- Use a detailed example Keras gridsearch.
- Understand how to return the best parameters.

Part 5: Deep Learning with a Financial Markets focus

Note, delegates will learn by coding themselves. We make use of extensive examples that relate to the financial markets, modelling, econometrics and factor models. You can customise the agenda to focus on alternative areas if required. The agenda below is based on using Tensorflow. Pytorch can be used as an alternative (or both).

1. From Machine Learning to Deep Learning.

Delegates will be able to:

- Differentiate between machine learning and deep learning.
- Provide the background to Tensorflow and Keras
- Understand how the biological neuron works.
- Understand how the artificial neuron works.
- Provide the big picture of neural networks.
- Describe hidden layers.
- Describe nodes and activation functions.
- Describe error functions.
- Provide an overview of back propagation.
- Describe weights and bias of a neuron and a neural network.
- Describe the learning rate.

2. Building an Ann.

Delegates will be able to:

- Provide an overview of gradient descent using Excel.
- Provide alternative notation of gradient descent and cost functions.
- Implement gradient descent in Excel and Python.
- Define and implement the MNIST dataset.
- Understand and implement Perceptrons in python.
- Use the step function in python.
- State and describe the problems with Perceptrons.
- Define and measure Neuron saturation.
- Describe different types of layers.
- Describe and implement weighted inputs and activation functions.
- Describe and implement the cost functions and optimizers.
- Create metrics for performance and the training loop.
- Use the sigmoid activation function with Keras.
- Use the tanh activation function with Keras.
- Use the Rectified linear units (RELU) with Keras.
- Implementing SHAP and XAI on ANN's.

3. Implementing a Multi-Layer Perceptron in Keras.

Delegates will be able to:

- Implement One hot encoding with Keras.
- Use Keras to evaluate the deep learning model.
- Explain local minimums versus global minimums.
- Implement the softmax activation function in the final layer with Keras.
- Define rules of thumb for layers, neurons and activations.
- Define rules of thumb for the learning rate, epochs and batch size.
- Explain the math of back propagation.
- Use the tensorboard dashboard.
- Implement an ANN Regression.

4. Optimizers

Delegates will be able to:

- Define and implement Stochastic Gradient Descent in Keras.
- Define and implement Momentum in Keras.
- Define and implement Nesterov momentum in Keras.
- Define and implement AdaGRad & AdaDelta in Keras.
- Define and implement RMS Prop in Keras.
- Define and implement Adaptive Moment Estimation – Adam in Keras

5. Unstable gradients, Batch Normalisation and Weight Initialization

Delegates will be able to:

- Explain and mitigate against Exploding gradients.
- Explain and mitigate against Vanishing gradients
- Understand the concept of Batch normalisation.
- Implement Batch normalisation in Keras.
- Use the Xavier Glorot distribution for weight initialization in Keras.

6. Convolutional Neural Networks and Recurrent Neural Networks

Delegates will be able to:

- Explain simple cells vs complex cells in the context of CNN's.
- Describe the growth and importance of CNN's.
- Implement and explain feature extraction.
- Implement and explain filtering.
- Implement and explain pooling and translation invariance.
- Implement and explain transfer Learning.
- Implement a CNN in Keras.
- Understand how the Sunspring RNN movie works.
- Define and implement long term short term memory units with Keras.
- Define and implement a RNN by looking at a time series of Naspers and mking predictions based on the history of prices.
- Implement a time series fundamental factor model using an RNN with Keras.

Part 6: Unsupervised Learning with Clustering and Principal Components analysis

Delegates will be able to:

- Describe different clustering methodologies.
- Create an elbow / scree plot with sklearn.
- Implement Hierarchical Clustering with sklearn.
- Implement Dendrograms with sklearn.
- Understand a PCA in Plain English.
- Define Eigenvalues and eigenvectors.
- Understand Covariance matrix in the context of a PCA.
- Generate the principal components using Sklearn and plot the component contributions.

Other optional modules include:

- **Time series analysis (recommended**). This section includes traditional methods and AI methods.
- **Reinforcement Learning.**
- **Anomaly detection and Regime Change** (including Markov chains).
- **Algo Trading & Portfolio Management.**
- **Automation & Webscraping etc**.
- **Implementing LLM's**.