*******Machine Learning with R and Python*******

1.

- Introduction to Supervised Learning
- Introduction to unsupervised learning
- Introduction to reinforcement learning
- Machine Learning versus Rule-based programming
- Understanding What Machine Learning can do using the Tasks Framework
- Creating Machine-Learning Models with Python and scikit learn.
- Types of datasets used in Machine Learning.
- Life Cycle of Machine Learning
- Dealing with Missing Values – An example
- Standardization and Normalization to Deal with Variables with Different Scales
- Types of scaling techniques
- Eliminating Duplicate Entries
- Learning Rules to Classify Objects?
- Understanding Logistic Regression
- Applying Logistic Regression to The Iris classification Task
- Closing Our First Machine Learning Pipeline with a Simple Model Evaluator
- Creating Formulas that predict the Future – A House Price Example
- Understanding Linear Regression
- Applying Linear Regression to the Boston House Price Task
- Evaluating Numerical Predictions with Least Squares
- Gradient Descent Algorithm
- Batch Gradient Descent
- Stochastic Gradient Descent algorithm
- Exploring Unsupervised Learning and Its Usefulness
- Finding Groups Automatically with k-means clustering
- Reducing The Number of variables in your data with PCA
- Smooth out your Histograms with kernel Density Estimation
- Decision Trees Classifier
- Decision Tree Regressor
- Random Forest Classifier
- Random Forest Regressor
- Automatic Feature Engineering with Support Vector Machines
- Deal with Nonlinear Relationships with Polynomial Regression
- Reduce the number of Learned Rules with Regularization

II.

- Using Feature Scaling to Standardize Data
- Implementing Feature Engineering with Logistic Regression
- Extracting Data with Feature Selection and Interaction

- **Combining all Together**
- **Build Model Based on Real-world Problems**
- **Support Vector machines**
- **Implementing kNN on the Data set**
- **Decision Tree as Predictive Model**
- **Dimensionality Reduction techniques**
- **Combining all Together**
- **Random Forest for Classification**
- **Gradient Boosting Trees and Bayes Optimization**
- **CatBoost to Handle Categorical Data**
- **Implement Blending**
- **Implement Stacking**
- **Memory-Based Collaborative Filtering**
- **Item-to-Item Recommendation with kNN**
- **Applying Matrix Factorization on Datasets**
- **Word batch for Real-world Problem**
- **Validation Dataset Tuning.**
- **Regularizing model to avoid over fitting**
- **Adversarial Validation**
- **Perform metric Selection on real Data.**
- **Tune a linear model to predict House prices**
- **Tune an SVM to predict a politician's Party Based on their Voting Record**

III.

- **Splitting your datasets into train, test and validate**
- **Persist Models by Saving Them to Disk**
- **Transform your variable length Features into One-Hot Vectors**
- **Finding the most important Features in your classifier**
- **Predicting Multiple Targets with the Same Dataset**
- **Retrieving the Best Estimators after Grid Search**
- **Extracting Decision Tree Rules from Scikit-learning**
- **Finding out which features are important in Random Forest Model**
- **Classifying with SVMs, when your data has unbalanced classes**
- **Computing True/False Positives/Negatives after in scikit-learn**
- **Labelling Dimensions with Original Feature Names after PCA**
- **Clustering Text Documents with Scikit-learn k-means**
- **Listing Word Frequency in a Corpus Using Only scikit-learn**
- **Polynomial Kernel Regression Using Pipelines**
- **Visualize outputs over two dimensions using Numpy's Meshgrid**
- **Drawing out a Decision Tree Trained in scikit-learn**
- **Clarify your Histogram by Labeling each Bin**
- **Centralizing Your Color legend when you have multiple subplots**

IV.

- **Programming with TENSORFLOW**
- **Implementation of all above models with TENSORFLOW**