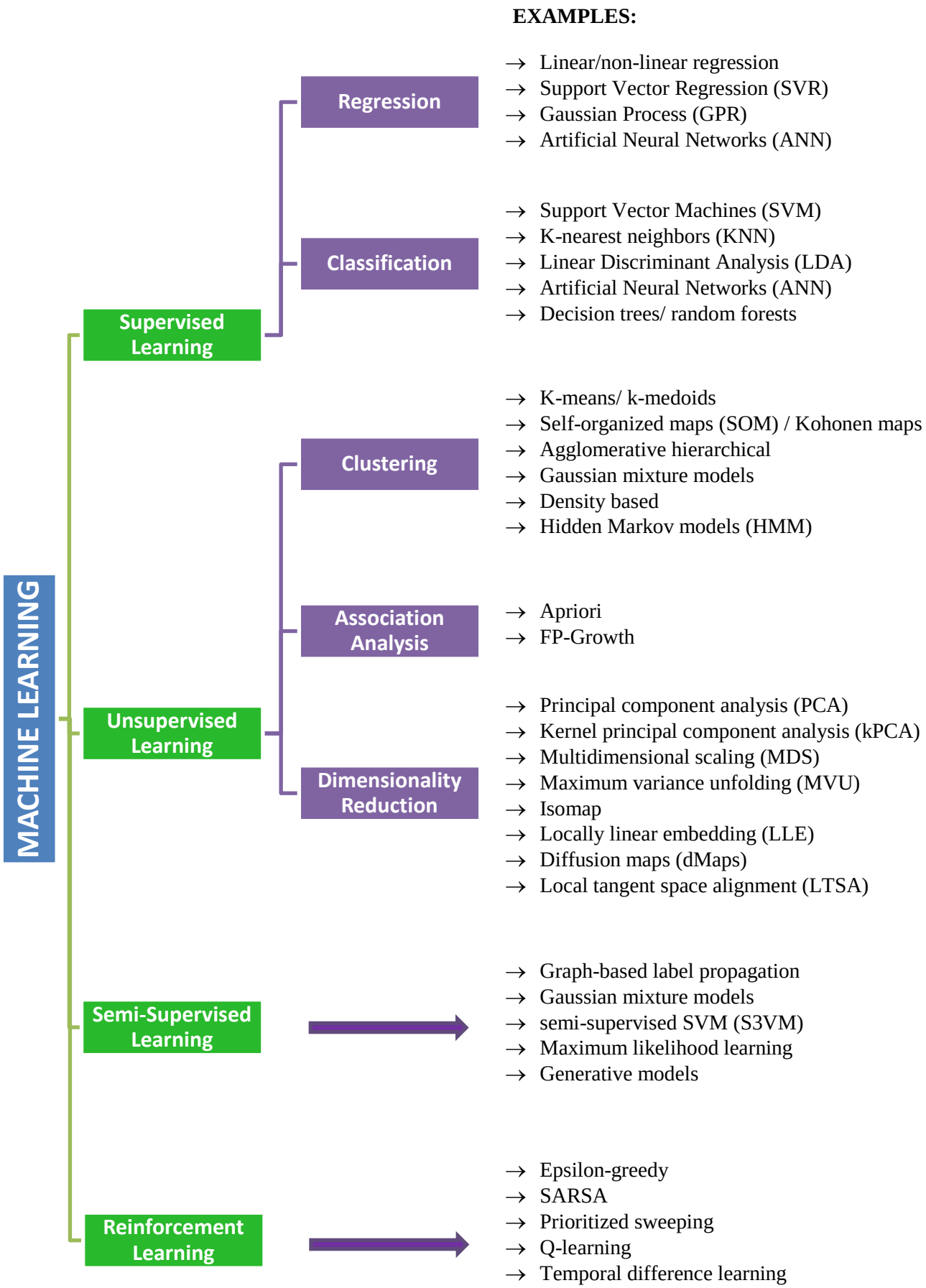
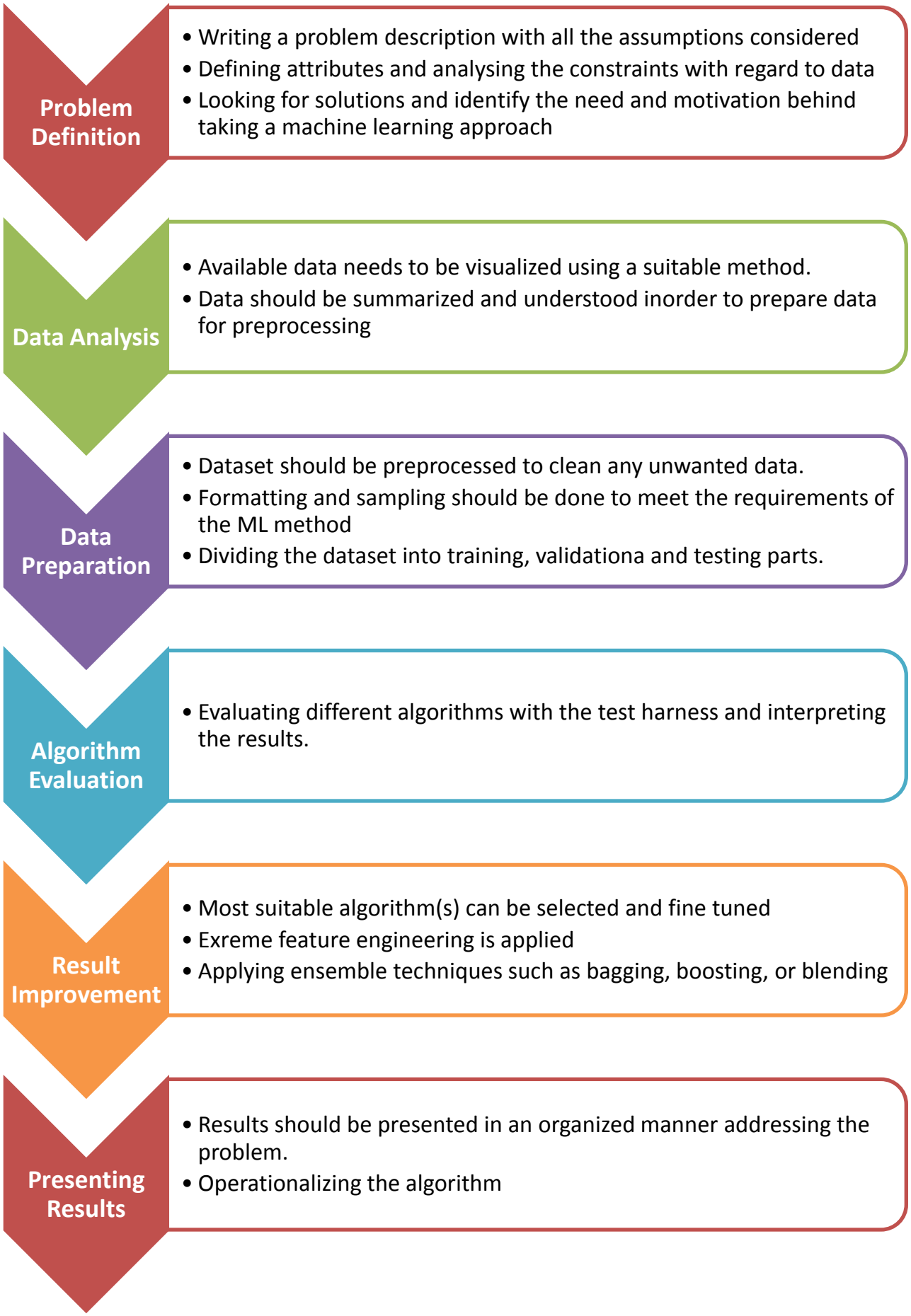
Question 01


MACHINE LEARNING
Supervised Learning
Regression
Classification
Unsupervised Learning
Clustering
Association Analysis
Dimensionality Reduction
Semi-Supervised Learning
Reinforcement Learning
EXAMPLES:
→ Linear/non-linear regression
→ Support Vector Regression (SVR)
→ Gaussian Process (GPR)
→ Artificial Neural Networks (ANN)
→ Support Vector Machines (SVM)
→ K-nearest neighbors (KNN)
→ Linear Discriminant Analysis (LDA)
→ Artificial Neural Networks (ANN)
→ Decision trees/ random forests
→ K-means/ k-medoids
→ Self-organized maps (SOM) / Kohonen maps
→ Agglomerative hierarchical
→ Gaussian mixture models
→ Density based
→ Hidden Markov models (HMM)
→ Apriori
→ FP-Growth
→ Principal component analysis (PCA)
→ Kernel principal component analysis (kPCA)
→ Multidimensional scaling (MDS)
→ Maximum variance unfolding (MVU)
→ Isomap
→ Locally linear embedding (LLE)
→ Diffusion maps (dMaps)
→ Local tangent space alignment (LTSA)
→ Graph-based label propagation
→ Gaussian mixture models
→ semi-supervised SVM (S3VM)
→ Maximum likelihood learning
→ Generative models
→ Epsilon-greedy
→ SARSA
→ Prioritized sweeping
→ Q-learning
→ Temporal difference learning
→ Markov Decision Process
→ Multi-armed bandits

Supervised learning	Model is trained by finding relationships between input data to their labels.
Unsupervised learning	Input data is not labeled. General structure/data geometry is extracted in this learning method.
Semi-supervised learning	Input data is a mixture of labeled and unlabeled data. So the accuracy of only using unsupervised data can be improved with this approach.
Reinforcement learning	Model is created by learning an optimal policy using the reward feedback.

Question 02


Problem Definition
• Writing a problem description with all the assumptions considered
• Defining attributes and analysing the constraints with regard to data
• Looking for solutions and identify the need and motivation behind taking a machine learning approach
Data Analysis
• Available data needs to be visualized using a suitable method.
• Data should be summarized and understood inorder to prepare data for preprocessing
Data Preparation
• Dataset should be preprocessed to clean any unwanted data.
• Formatting and sampling should be done to meet the requirements of the ML method
• Dividing the dataset into training, validationa and testing parts.
Algorithm Evaluation
• Evaluating different algorithms with the test harness and interpreting the results.
Result Improvement
• Most suitable algorithm(s) can be selected and fine tuned
• Exreme feature engineering is applied
• Applying ensemble techniques such as bagging, boosting, or blending
Presenting Results
• Results should be presented in an organized manner addressing the problem.
• Operationalizing the algorithm

Question 03

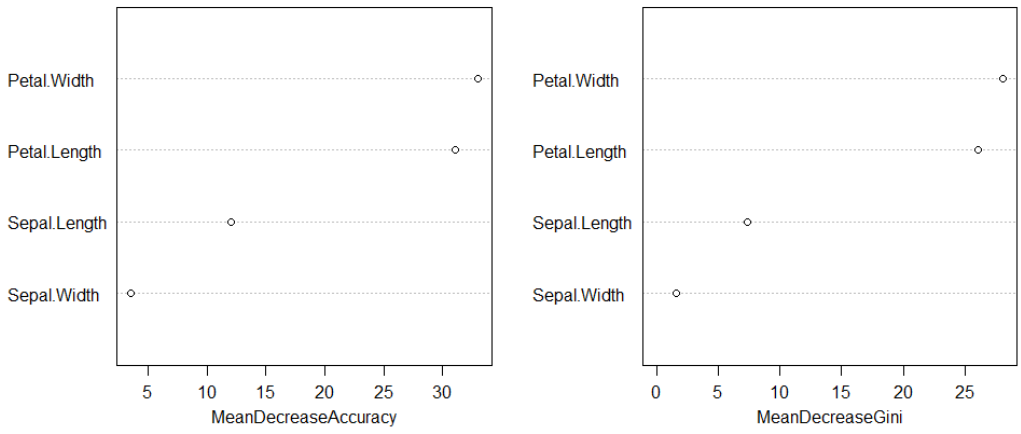
Random forest classifier is an improved version of the decision tree classifier. In this method the attributes for the tree nodes are selected randomly rather than getting them by calculating the impurity. Also since decision tree are highly sensitive to the dataset, random forest classifiers exploit this characteristics to give a more accurate result with the out-of-bag data. In random forest classifiers there are a large number of uncorrelated trees that work together after getting trained using bootstrapped datasets. The aggregated result from each model is then taken as the output of the whole model.

However random forest classifiers use two main methods to calculate variable importance. They are,

1. Mean decrease in accuracy
2. Mean decrease in Gini impurity

Let's consider the Iris data set. In order to find the most suitable variables in this dataset we could use the random forest classifier. Both of the methods yield similar results in this case. Difference in values is actually not important. Only the relative values matter. For example, from the following figures we can say that sepal.length is 7 times more important than sepal.width.

```
> varImpPlot(rf, main = "Iris data set variable Importance")
> importance(rf, type=1)
             MeanDecreaseAccuracy
Sepal.Length            12.000162
Sepal.Width              3.541789
Petal.Length            31.131702
Petal.Width             33.003290
> importance(rf, type=2)
             MeanDecreaseGini
Sepal.Length         7.402288
Sepal.Width          1.637228
Petal.Length        26.034754
Petal.Width         28.089415
>
```



1. Permutation/Accuracy based importance

   The prediction accuracy of out-of-bag samples are measured as the first step. Then a single variable is selected and the values of this variable are randomly shuffled keeping everything else the same. Then

the prediction accuracy is calculated just as before. Then the mean decrease in accuracy across all the trees is recorded. Larger the decrease in accuracy, higher the importance of the variable.

2. Gini impurity based importance/Mean decrease in impurity(MDI)
   Mean Decrease in Gini is the average (mean) of a variable's total decrease in node impurity, weighted by the proportion of samples reaching that node in each individual decision tree in the random forest. This is effectively a measure of how important a variable is for estimating the value of the target variable across all of the trees that make up the forest. A higher Mean Decrease in Gini indicates higher variable importance.

There are few other methods used in importance calculation as well such as drop-column importance but the most common ones are described above. However there can be slight differences in ways they are implemented in different data science packages in different tools.

Question 04

Problem Statement

The problem is to create a regression model for the variation of salary with the year of experience of employees in company.

The dataset was divided into two in order to test the model and train the model. First the gradient and intercept for the linear best suited function is calculated. This was calculated as follows.

$$Gradient = \frac{Covariance}{Variance}$$

$$Intercept = \bar{y} - Gradient * \bar{x}$$

Then after that the real years of experience is taken and using the regression coefficients we calculated, the predictions for the salary were derived.
Finally the root mean square error between the test salary and predicted salary values are calculated as an evaluation criteria.

Optimization is done by using vector functions in python numpy library. Also we can improve the accuracy of the model by implementing other methods beforehand applying regression. For example, we can select the most important variables using a method such as Random Forest Classification, and then apply regression to the best two variables.

If the dataset has a clear demarcation of categories, we can use the simple linear regression model as a simple linear classifier. In the same way we could calculate the gradient and intercept of a linear function taking suitable features for independent and dependent variables. But rather than giving a relationship between the two variables at this moment it would give a clear boundary between two categories. But this is rarely accurate. So there are other more suitable ML algorithms to do classifications. We can predict the category of the values based on the side that it belongs to considering the features we have used for the axes.

Relevant code can be found in the name “ReggressionModel.py” and the relevant data is found under “Salary_Data.csv” in the same folder.