**1.1. Main Points Covered in Week 7 and 8**

**Week 7 – Support Vector Machines (SVM):**

* Introduction to SVM for linearly separable and non-separable data.
* Kernel trick and non-linear SVM for handling complex decision boundaries.
* Support Vector Regression (SVR) for regression tasks.
* Statistical learning theory of SVM (VC dimension, margin theory).
* Multi-class classification methods in SVM (e.g., one-vs-rest, one-vs-one).
* Practical implementation of SVM in Python using **linear, polynomial, and RBF kernels**.

**Week 8 – KNN and Decision Trees:**

* Overview of the K-Nearest Neighbors (KNN) algorithm and its variants.
* Theory of KNN and importance of choosing the best number of neighbors (K).
* Decision trees: structure, splitting rules, entropy, and information gain.
* Regression trees and classification trees.
* Algorithms for building decision trees (e.g., CART, ID3, C4.5).
* Model complexity, pruning techniques, and bias-variance trade-off.
* Advantages and disadvantages of decision trees.
* Practical implementation of **KNN and decision trees in Python**.

**1.2. Summary of Reading List (External Resources, etc.)**

* **Textbook chapters**: Machine Learning texts often cover SVM (kernels, optimization) and decision trees (splitting, pruning).
* **Websites**: Scikit-learn documentation (sklearn.svm, sklearn.neighbors, sklearn.tree) gives practical code examples.
* **External tutorials**: Towards Data Science, Analytics Vidhya, and Medium blogs provide intuitive explanations and visualization for SVM and decision trees.
* **Code libraries**:
  + **Scikit-learn** → SVC, SVR, KNeighborsClassifier, DecisionTreeClassifier, DecisionTreeRegressor.
  + **NumPy & Pandas** for handling datasets.
  + **Matplotlib/Seaborn** for visualization of decision boundaries and model results.

**1.3. Reflection on Knowledge Gained**

From these two weeks, I gained deeper insights into **classification and regression algorithms**.

* SVM taught me how **maximizing the margin** improves generalization and how the **kernel trick** enables handling non-linear data. I also understood that SVM is powerful but can be computationally expensive with large datasets.
* With KNN, I learned its **simplicity and intuition** but also its sensitivity to the choice of **K** and computational inefficiency in large datasets.
* Decision trees showed me how machine learning models can be **interpretable**, but also prone to **overfitting** unless pruning is applied.
* Implementing these in Python reinforced the theory and showed me how **hyperparameters (kernel choice, depth, neighbors)** directly affect performance.

Overall, this week improved my understanding of the **balance between model complexity, interpretability, and generalization** in machine learning. A screenshot of a quiz

AI-generated content may be incorrect.A screenshot of a quiz

AI-generated content may be incorrect.