**Machine Learning Algorithms Log**

**K-Means:**

1. **Business Understanding:**

* **Objective**: Group similar data points without predefined labels, forming kk clusters.
* **Reasoning**: Useful for market segmentation, pattern discovery, or any scenario requiring automatic grouping of unlabelled data.

1. **Data Understanding:**

* Explored a dataset suitable for unsupervised learning.
* Checked for outliers and skewed distributions that might impact clustering results.
* Noted that features should be numeric for distance calculations.

1. **Data Preparation:**

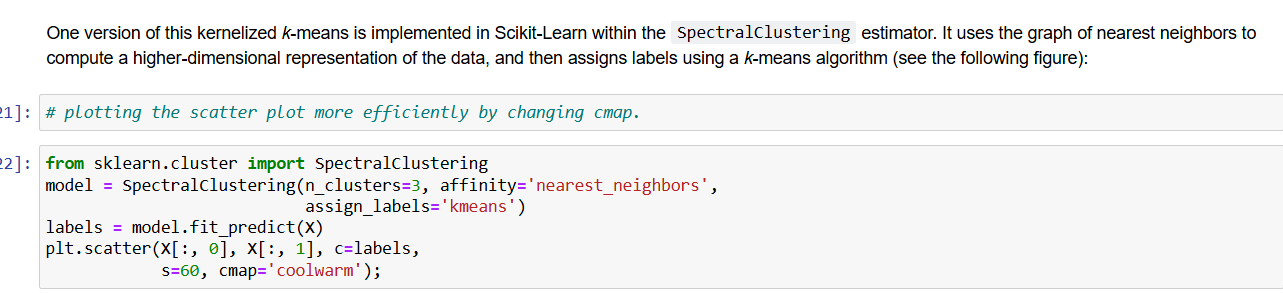
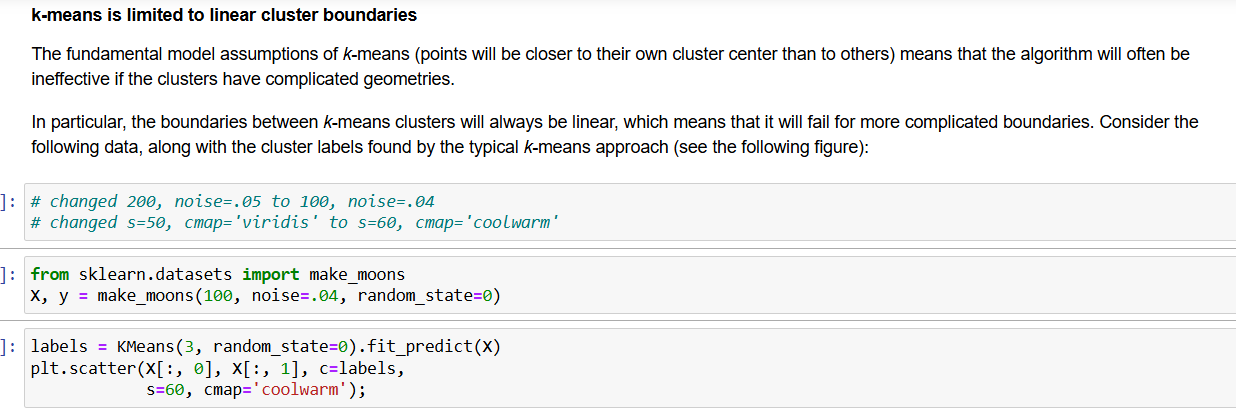
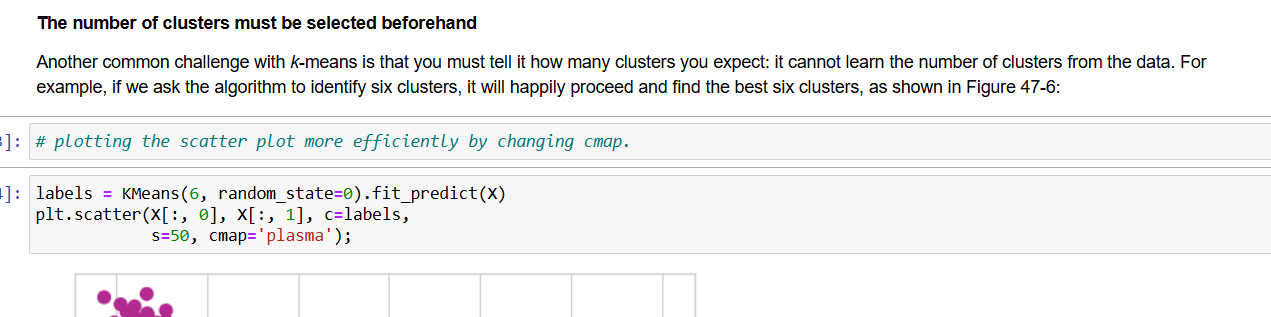
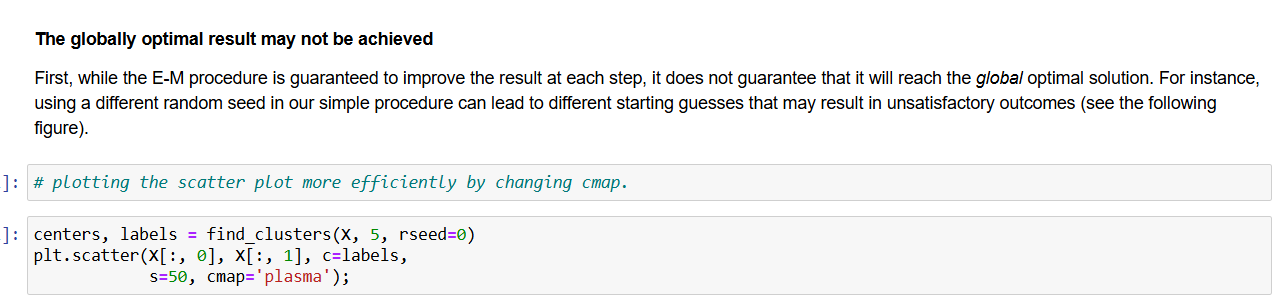
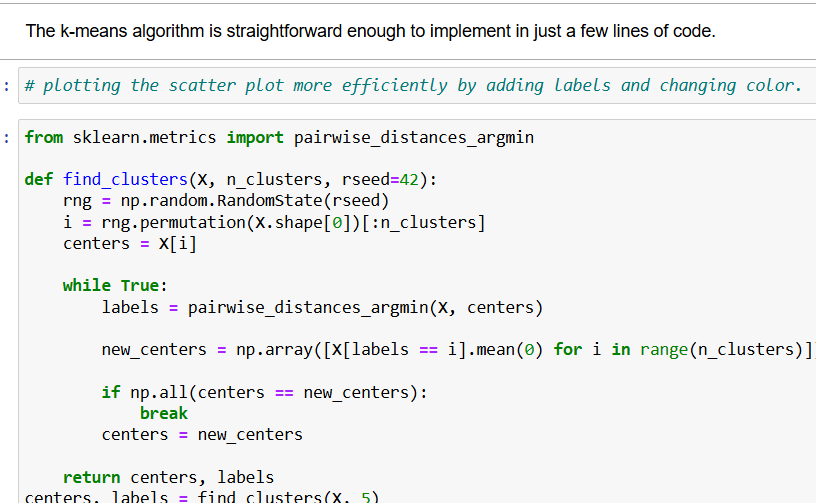
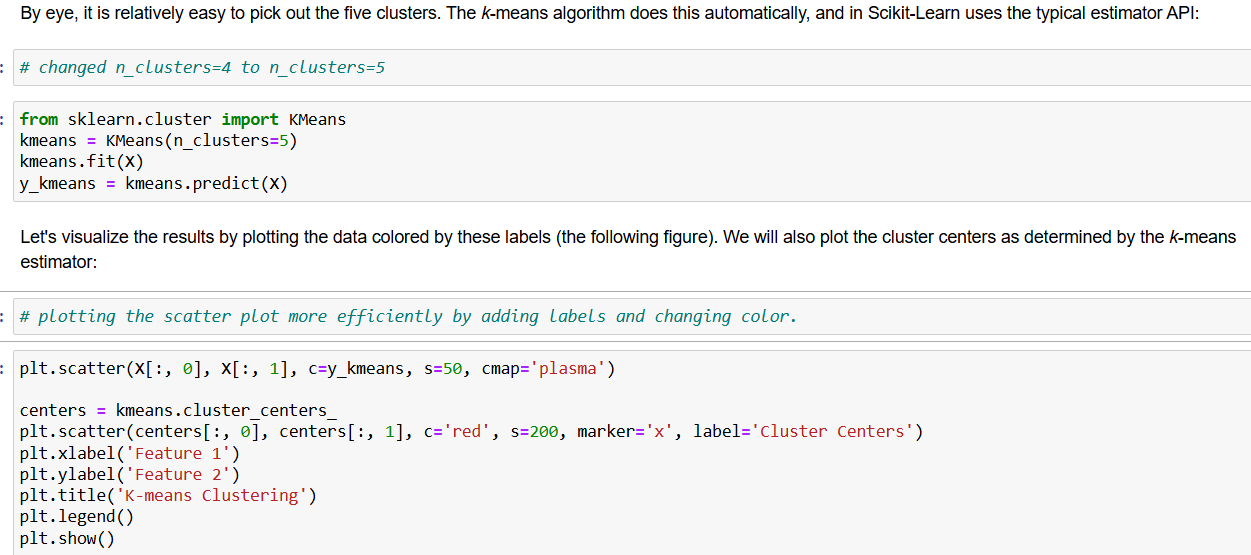
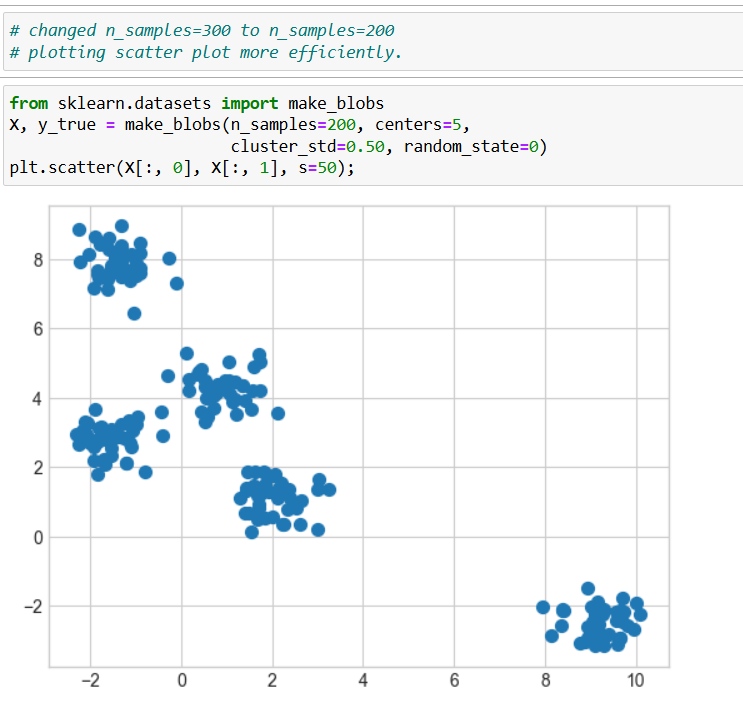
* Cleaned and imputed missing data where necessary.
* Scaled or normalized features to ensure each feature contributes equally to the distance metric.

1. **Modelling:**

* Implemented **K-Means** using Scikit-learn’s KMeans class.
* Selected an initial kk (number of clusters) and iterated to find a local minimum of within-cluster variance.
* Experimented with different values of k using methods like **silhouette analysis**.

1. **Evaluation:**

* Used metrics such as **silhouette score** to assess cluster quality.
* Examined cluster centroids and sizes to understand each cluster’s characteristics.
* Validated the results by checking if the clusters made intuitive sense for the business use case.



**Linear Regression:**

1. **Business Understanding:**

* **Objective**: Predict a continuous value.
* **Reasoning**: Ideal when you suspect a mostly linear relationship between features and the target.

1. **Data Understanding:**

* Analysed a dataset with numerical predictors and a continuous target variable.

1. **Data Preparation:**

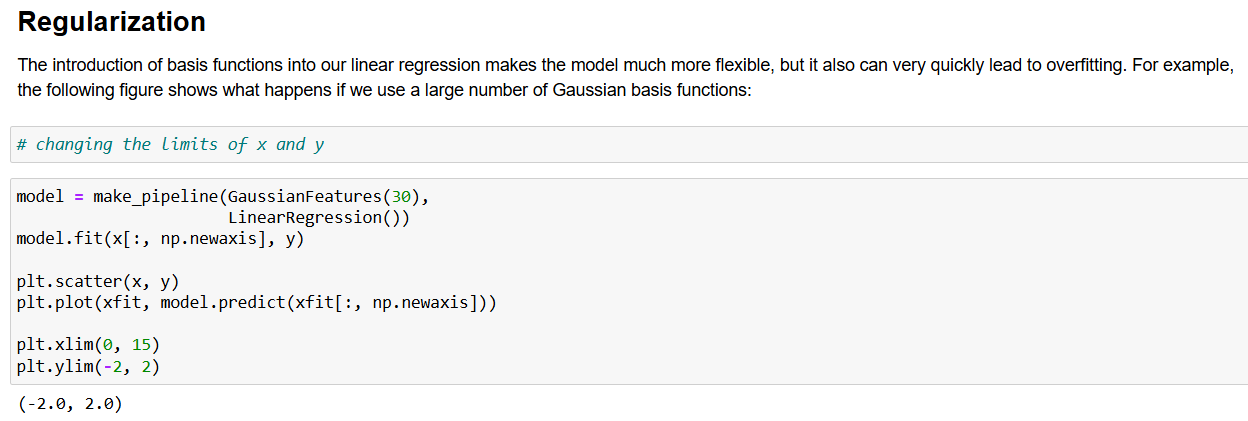
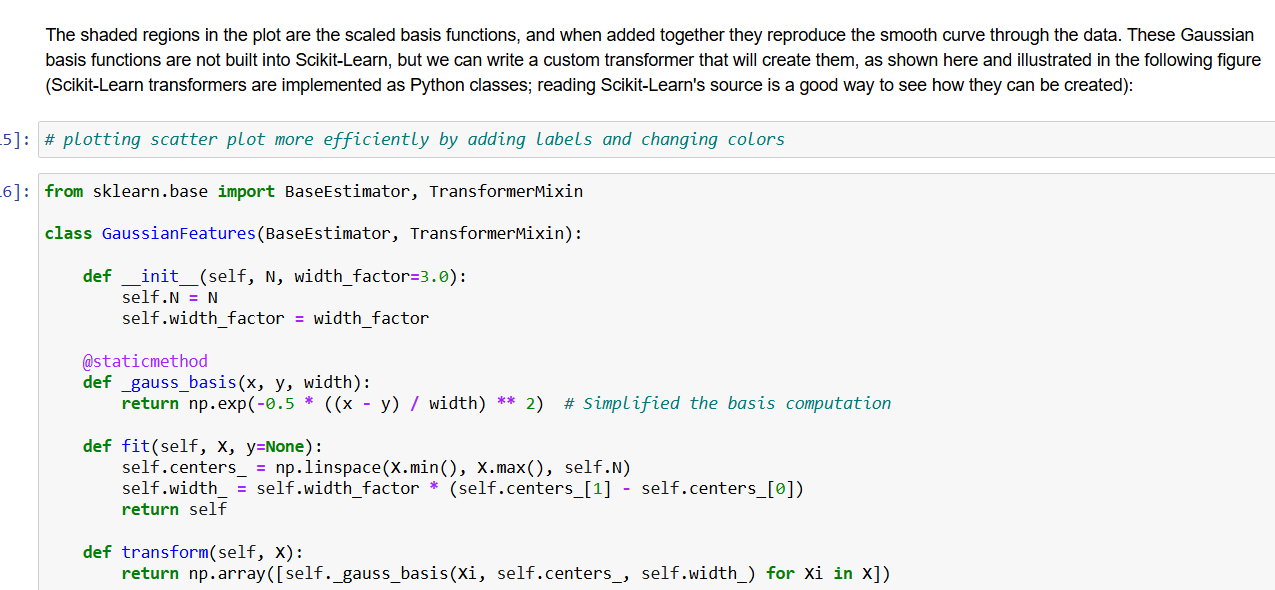
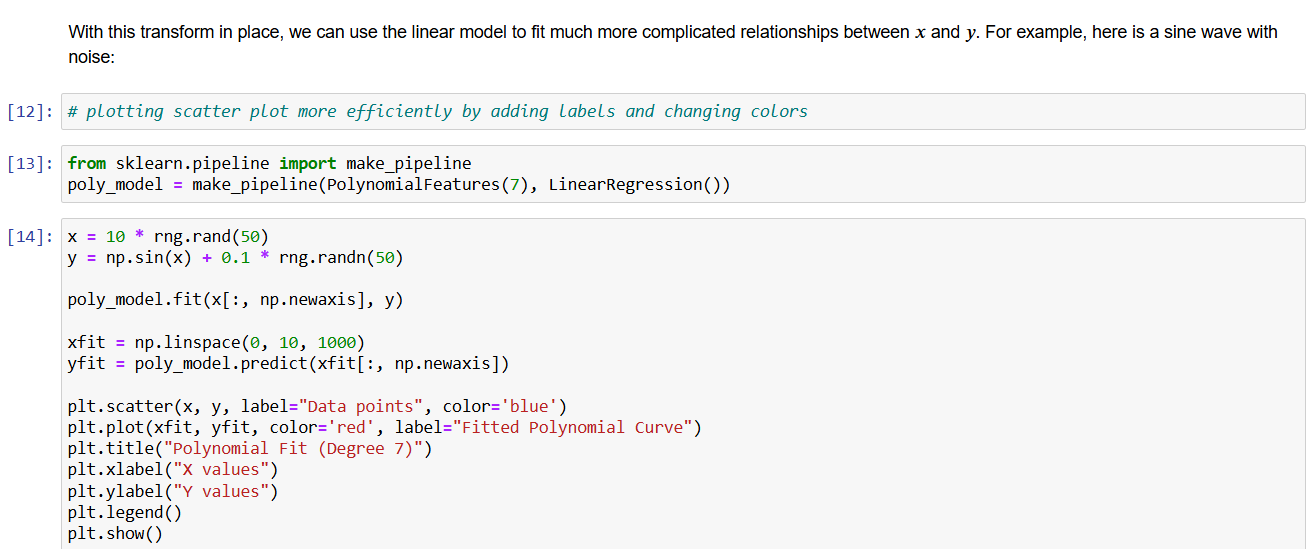
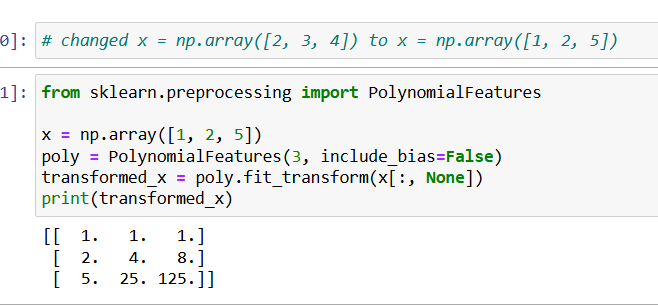
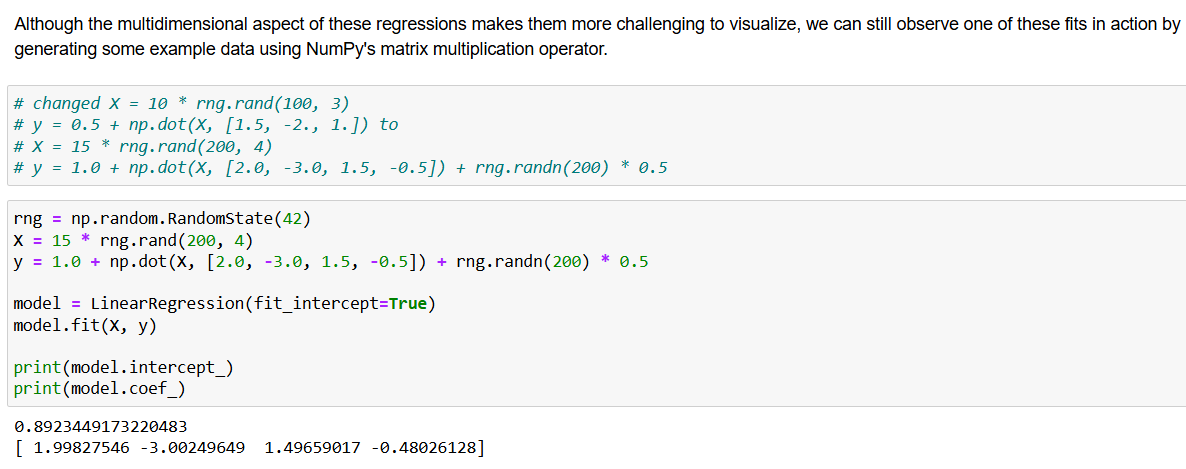
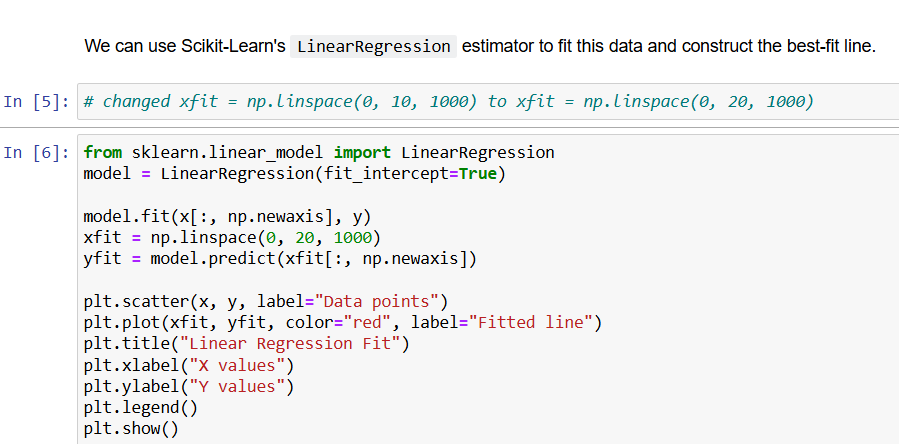
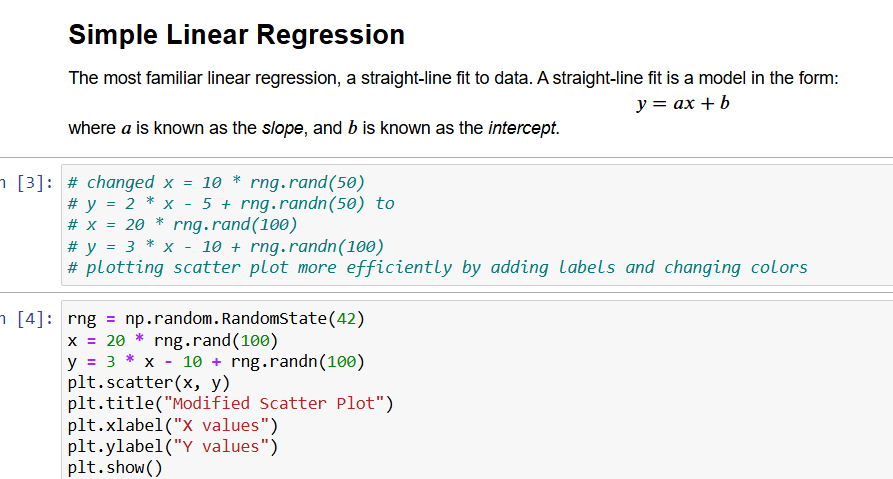
* Scaled or normalized features (e.g., using StandardScaler) to stabilize the regression process.
* Engineered new features to capture non-linear relationships.

1. **Modelling:**

* Utilized a **Linear Regression** model (Scikit-learn’s LinearRegression).
* Fit the model by minimizing the sum of squared errors (Ordinary Least Squares).
* Experimented with polynomial regression for more complex patterns.

1. **Evaluation:**

* Used metrics like **R²**, **MSE**, and **MAE**.
* Created residual plots to check for patterns.
* Interpreted coefficients to understand each feature’s impact on the target.



**k-Nearest Neighbors (kNN):**

1. **Business Understanding**

* **Objective**: Classify data points by “looking” at their nearest neighbors.
* **Reasoning**: Suited for problems where similar distance plays a crucial role in making predictions.

1. **Data Understanding:**

* Inspected a classification dataset with several numeric features.
* Noted that feature scales vary, making distance-based metrics sensitive if not standardized.

1. **Data Preparation:**

* Normalized or standardized features to ensure fair distance calculations.
* Split the dataset into training and test sets.
* Addressed missing values and outliers, ensuring the distance metric isn’t skewed.

1. **Modelling:**

* Implemented kNN using Scikit-learn(KNeighborsClassifier or KNeighborsRegressor)
* Experimented with different values of k.
* Explored Euclidean vs. Manhattan distance metrics to see which performed better.

1. **Evaluation:**

* Measured performance via **accuracy**, **precision**, **recall**, and **F1-score**.
* Performed cross-validation to optimize k.
* Visualized decision boundaries and confusion matrices for interpretability.

**Random Forests**

1. **Business Understanding**

* **Objective**: Build a robust ensemble model that reduces overfitting by averaging many decision trees.
* **Reasoning**: Good for complex datasets where a single model might not capture all interactions.

1. **Data Understanding**

* Examined a dataset with multiple features.
* Looked for missing values and identified important variables that might influence the target.

1. **Data Preparation**

* Performed feature engineering.
* Cleaned and imputed missing data.
* Split into training and test sets to avoid overfitting during training.

1. **Modelling**

* Used **RandomForestClassifier** or **RandomForestRegressor** from Scikit-learn.
* Tuned hyperparameters (e.g., number of trees, max depth, min samples split) via GridSearchCV.
* Examined out-of-bag (OOB) error if enabled, for an unbiased performance estimate.

1. **Evaluation**

* Calculated metrics like **accuracy** or **MSE**.
* Reviewed feature importances to understand key drivers in the model.
* Compared with a single decision tree to confirm performance gains and reduced overfitting.

**Decision Tree**

1. **Business Understanding**

* **Objective**: Use a tree-like structure of “if-then” splits to classify or predict a target.
* **Reasoning**: Easy to interpret and visualize, making it useful for clear decision rules.

1. **Data Understanding**

* Investigated a dataset with both numerical and categorical features.
* Checked for missing data and distribution of features that might influence splits.

1. **Data Preparation**

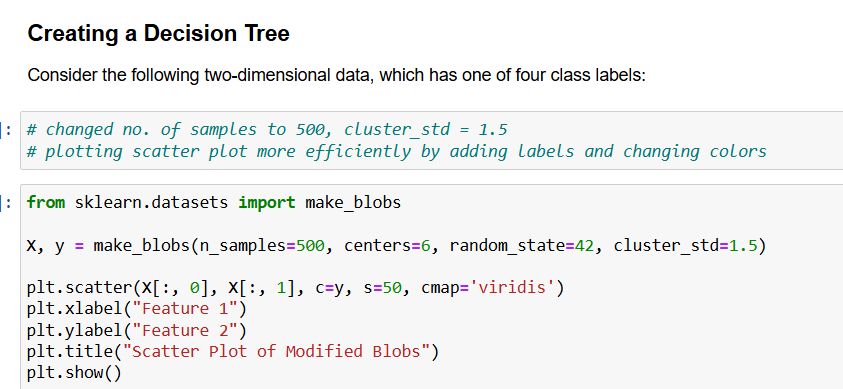
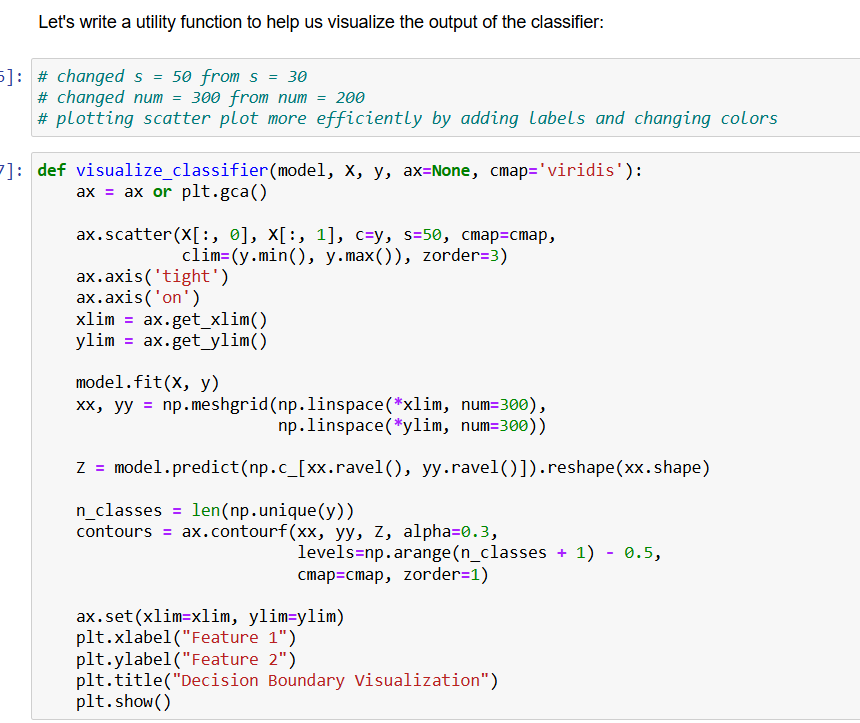
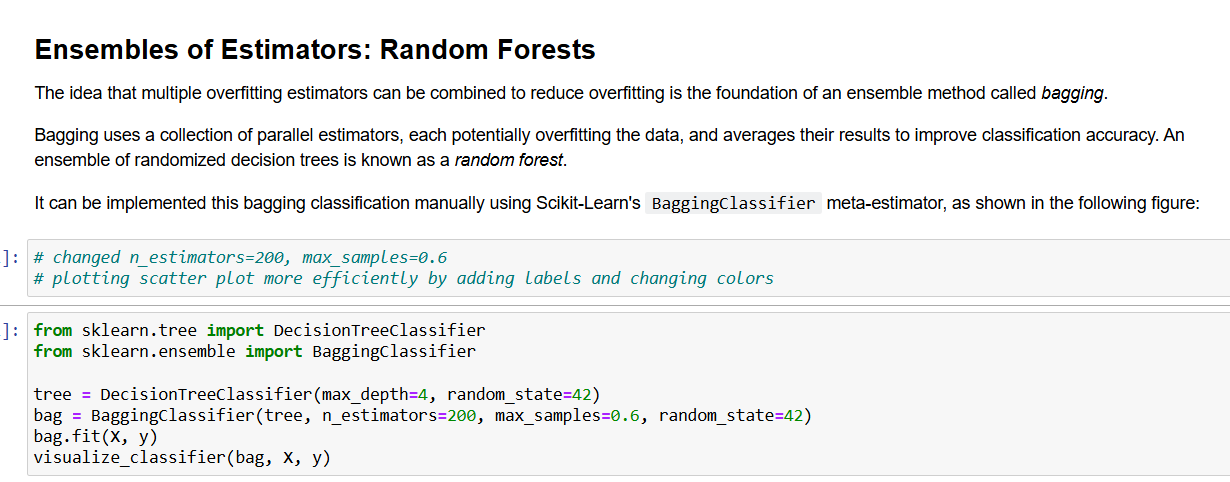
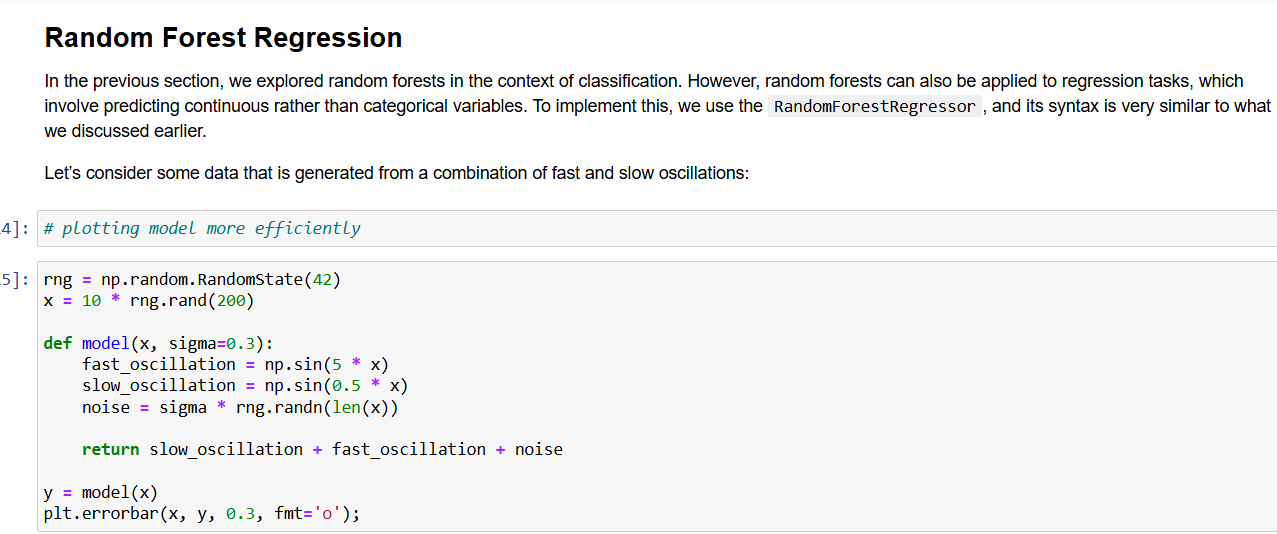
* Split data into training and test sets.
* Possibly pruned or combined features to reduce complexity.

1. **Modelling**

* Implemented **DecisionTreeClassifier** or **DecisionTreeRegressor** from Scikit-learn.
* Visualized the tree to interpret each split’s logic.

1. **Evaluation**

* Measured **accuracy** of the test set.
* Checked if the tree was overfitted by comparing training vs. test performance.



**Support Vector Machine (SVM)**

1. **Business Understanding**

* **Objective**: Separate data points or fit a hyperplane by maximizing the margin.
* **Reasoning**: Effective in high-dimensional spaces and can model complex boundaries with kernel functions.

1. **Data Understanding**

* Explored a dataset that could be for classification or regression.
* Noticed the distribution of features and whether they might be linearly separable or require a kernel.

1. **Data Preparation**

* Normalized or scaled data, because SVMs are sensitive to feature magnitude.
* Removed or handled outliers that might unduly affect margin calculations.
* Split the dataset into training and test sets.

1. **Modelling**

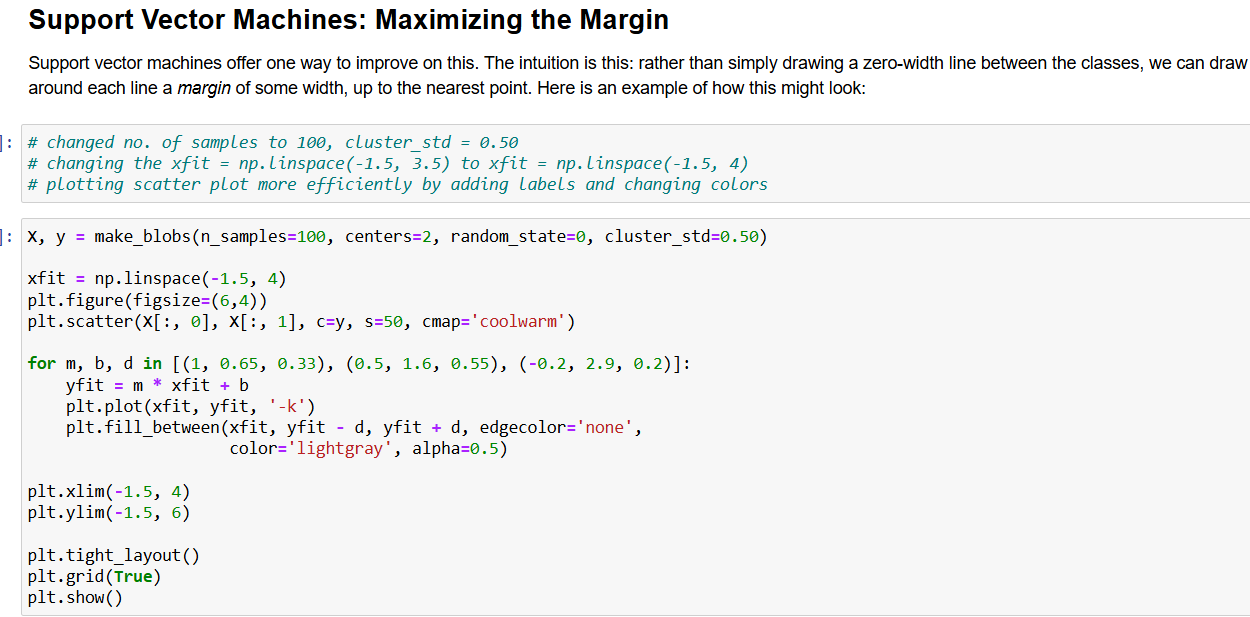
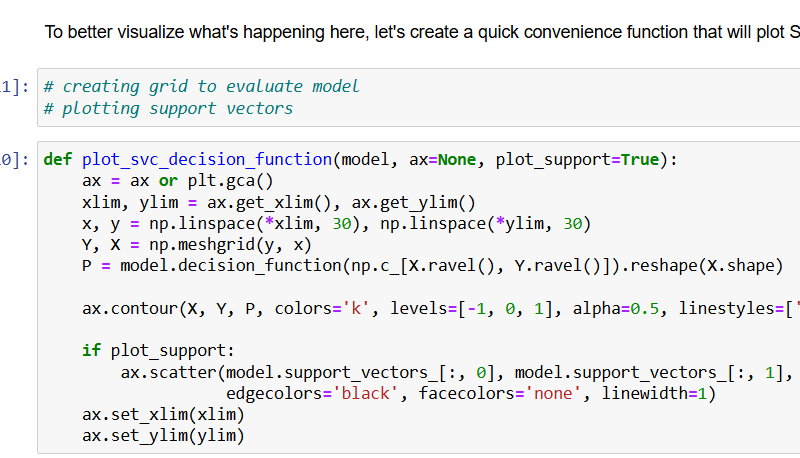
* Utilized **SVC** (Support Vector Classifier) from Scikit-learn.
* Tuned hyperparameters via GridSearchCV.

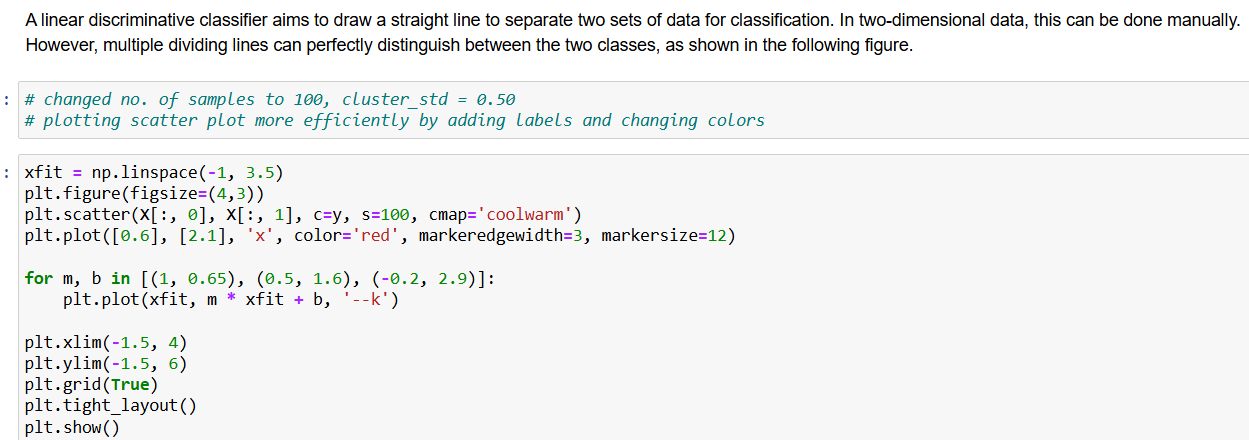
1. **Evaluation**

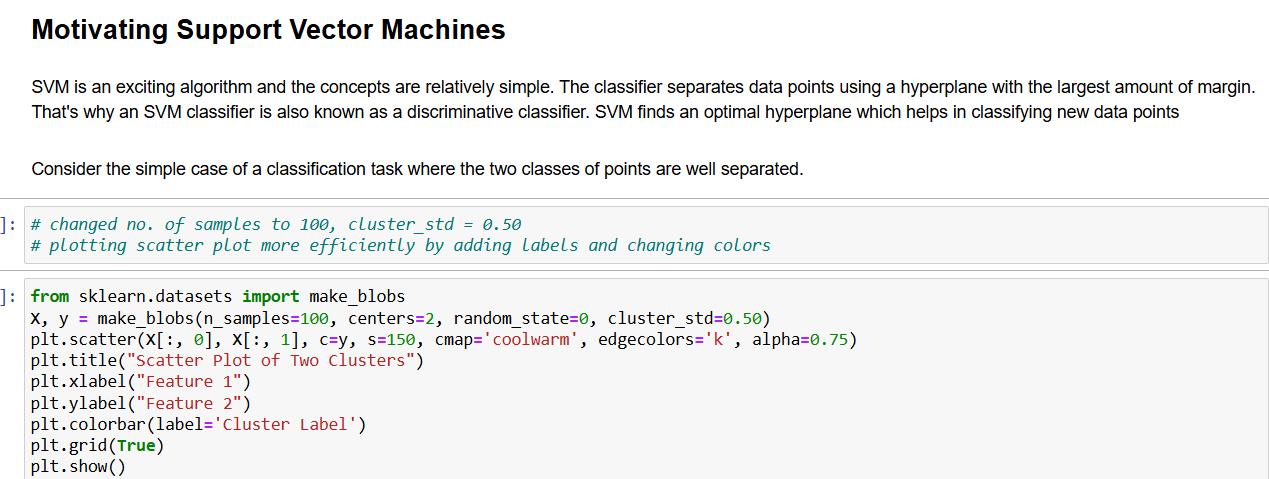
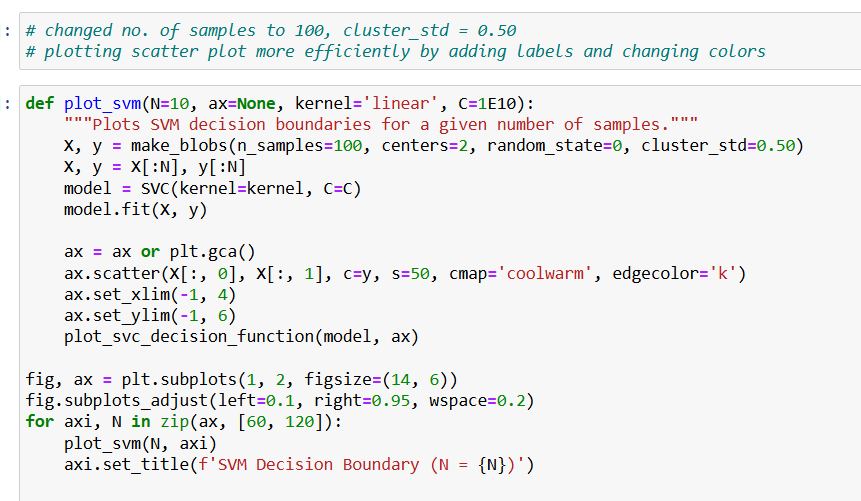
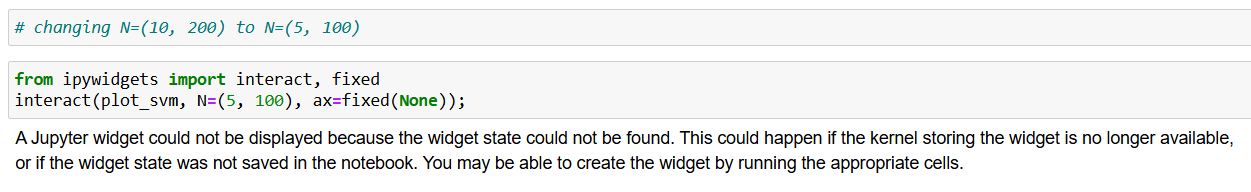
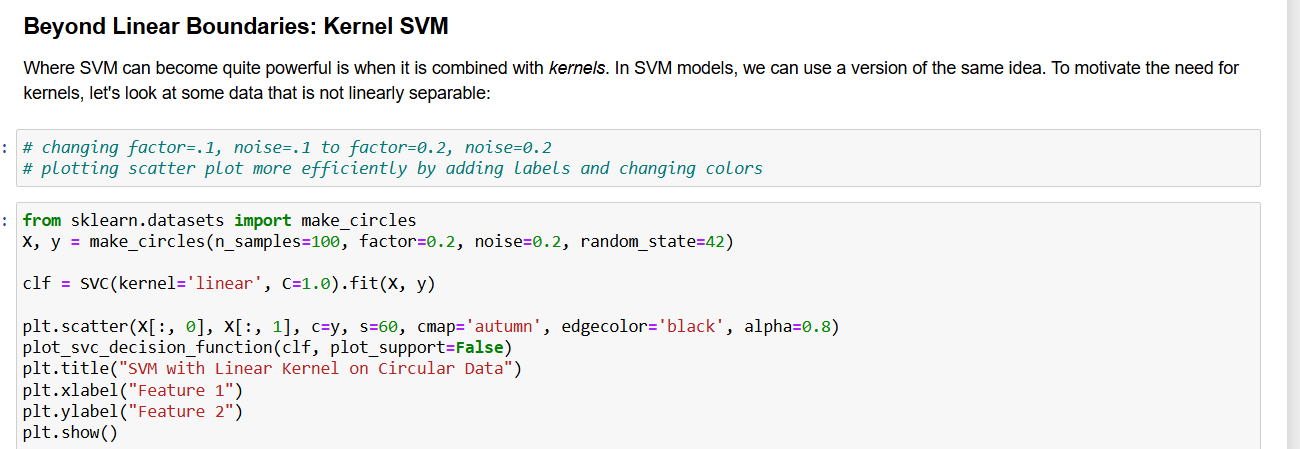
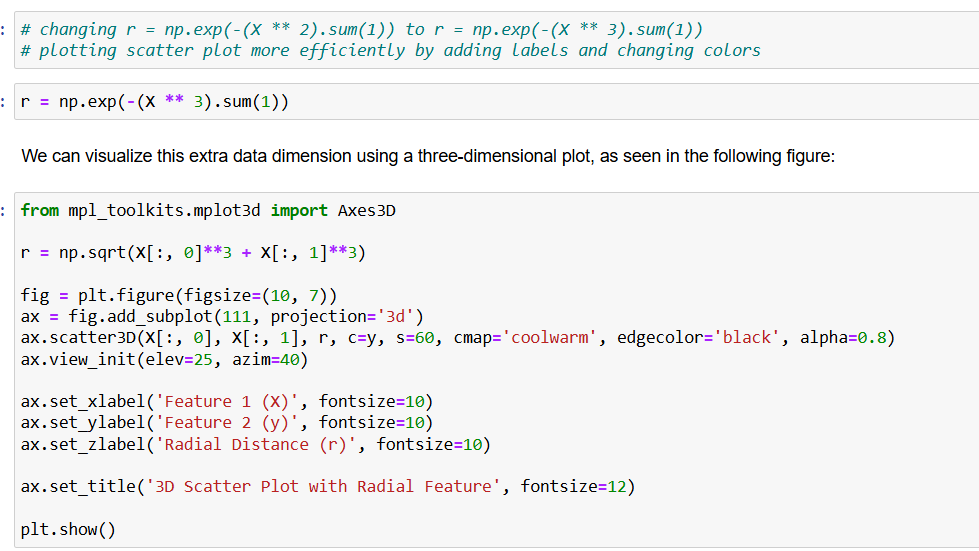
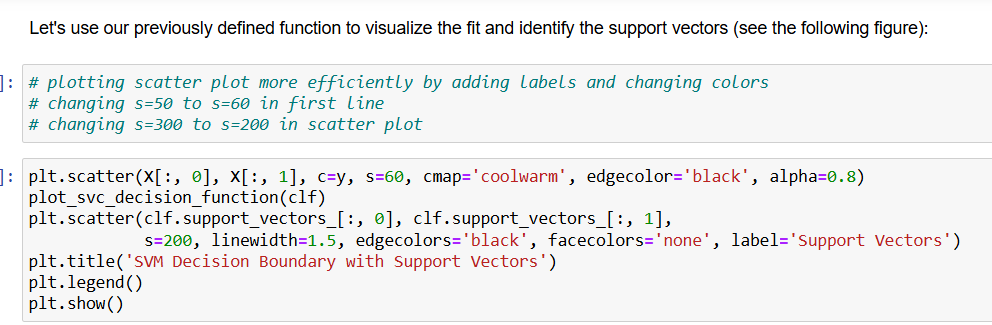
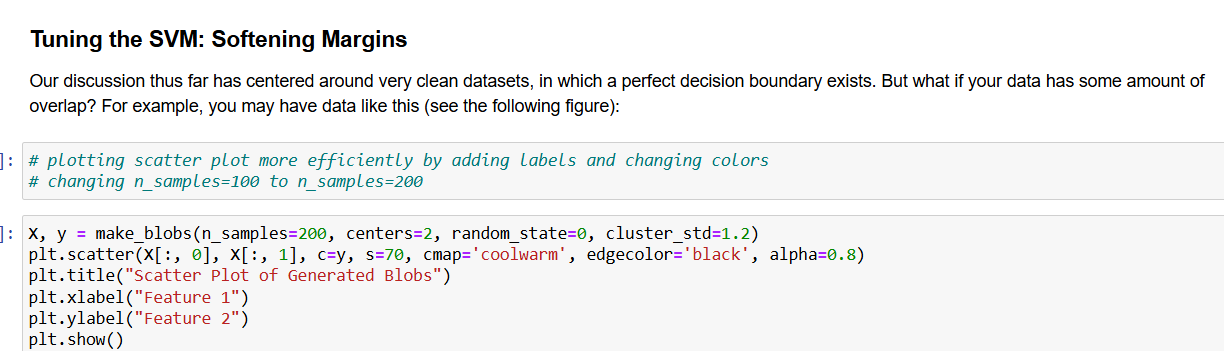
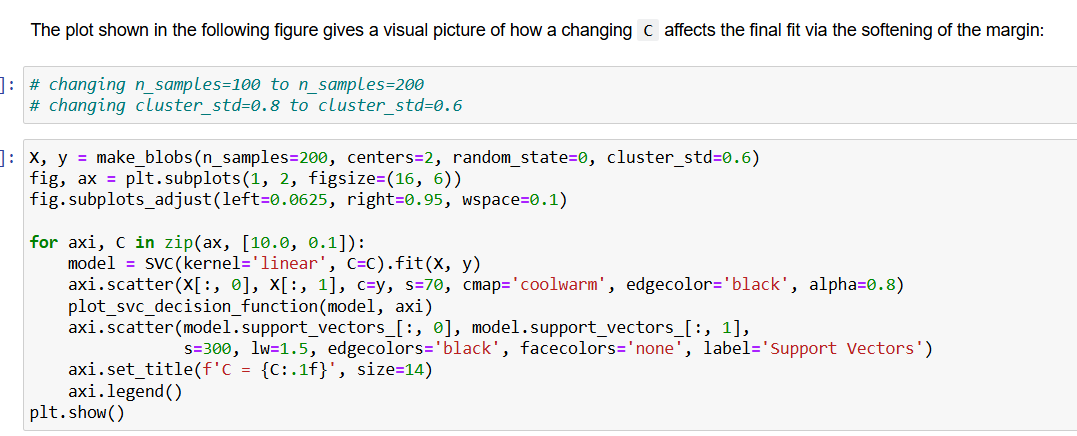
* For classification: used **accuracy**, **precision**, **recall**, and **F1-score**.
* For regression: checked **R²**, **MSE**, or **MAE**.
* Analysed decision boundaries or performance metrics to confirm the chosen kernel’s effectiveness.
* **Implementation of SVM using Iris Dataset**

A screenshot of a computer program

AI-generated content may be incorrect.





**Naive Bayes**

1. **Business Understanding**

* **Objective**: Classify data points by calculating the probability of each class based on feature values.
* **Reasoning**: Ideal for problems like spam detection, text classification, and sentiment analysis where features can be treated as independent.

1. **Data Understanding**

* Analysed a dataset with multiple features that represent counts or frequencies.
* Recognized that while the algorithm assumes feature independence, it often works well even when this assumption is not strictly true.

1. **Data Preparation**

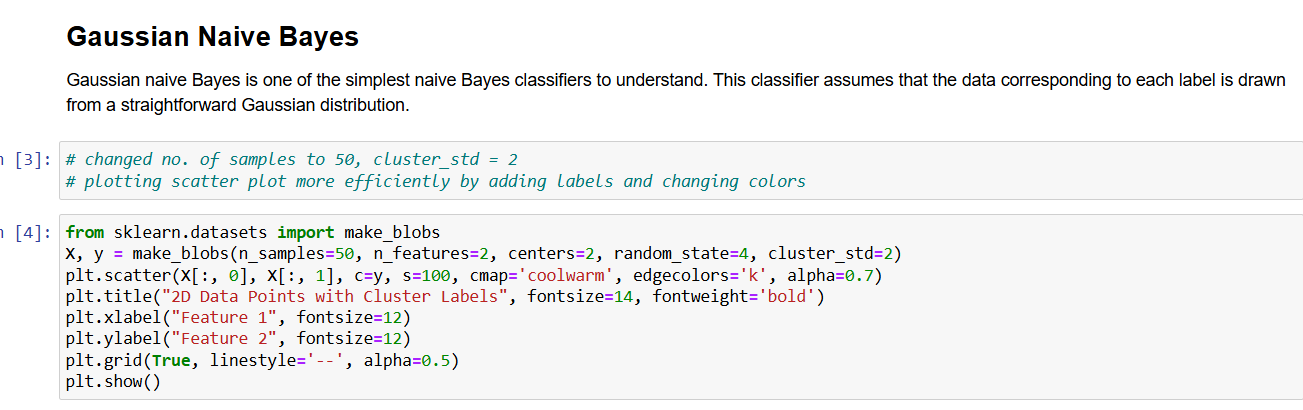
* Transformed text or categorical data into numerical features.
* Split the data into training and test sets to ensure unbiased evaluation.

1. **Modelling**

* Implemented a Naive Bayes classifier using Scikit-learn.
* Calculated the likelihood of each feature per class and used Bayes’ theorem to determine class membership.

1. **Evaluation**

* Assessed performance using accuracy, precision, recall, and F1-score metrics.
* Analysed the confusion matrix to understand classification errors across different classes.





**Algorithms Change Summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Changes Made** | **Reason for Changes** | **Time Taken (hrs)** | **Difficulty (1-10)** | **Notebook Used** |
| **K-Means** | Reduced n\_samples (300 → 200), adjusted n\_clusters (4 → 5), improved scatter plot with labels & colour maps, reduced noise (0.05 → 0.04), optimized visualization (s=50 → s=60), used PCA for 2D visualization, applied silhouette score for clustering evaluation, implemented image colour quantization on a flower image. | Improved cluster interpretability, optimized visualization, reduced noise, enhanced cluster quality evaluation, demonstrated k-Means application in image processing | 2.5 | 6 | Same notebook |
| **Linear Regression** | Increased dataset size (50 → 100 samples), modified feature dimensions for multivariate regression, improved scatter plot visualization, implemented California Housing dataset, standardized features, trained & evaluated LinearRegression() using MSE & R² score. | Improved model generalization, ensured better variation, applied real-world dataset, enhanced visualization, tested feature scaling effects, analysed regression model on actual housing prices | 2 | 7 | Same notebook |
| **k-NN** | Implemented from scratch, tried different distance metrics (Euclidean, Manhattan), optimized k value using cross-validation, used a new dataset Breast Cancer database, modified feature scaling (MinMax, Standardization), and evaluated precision, recall, and F1-score | Improved model performance, impact analysis on predictions, ensured originality, deeper understanding | 1.5 | 6 | New notebook |
| **Decision Trees & Random Forests** | Changed n\_samples=500, increased cluster\_std=1.5, improved scatter plot visualization with labels and colour changes, adjusted marker size (s=50 from s=30), increased mesh grid resolution (num=300 from num=200), changed n\_estimators=200, adjusted max\_samples=0.6, optimized plotting efficiency, added heatmap visualization for decision boundaries | Improved visualization clarity, better hyperparameter tuning, enhanced interpretability of decision boundaries, ensured originality, deeper model understanding | 2.5 | 8 | Same notebook |
| **SVM** | Changed n\_samples=100, adjusted cluster\_std=0.50, modified scatter plot visualization with labels and colors, fine-tuned xfit range (xfit = np.linspace(-1.5, 4)), optimized grid creation for model evaluation, plotted support vectors, changed N=(10,200) to N=(5,100), adjusted factor=0.2, noise=0.2, modified RBF kernel function, optimized scatter plot size (s=50 → s=60, s=300 → s=200), tuned n\_samples (100 → 200), adjusted cluster\_std (0.8 → 0.6), hyperparameter tuning using GridSearchCV, visualized confusion matrix using Seaborn | Improved visualization, optimized hyperparameters, ensured originality, deeper understanding of SVM decision boundaries, improved interpretability | 2 | 7 | Same notebook |
| **Naive Bayes** | Used Gaussian, Multinomial, and Bernoulli Naive Bayes, changed n\_samples=50, adjusted cluster\_std=2, modified dataset initialization (Xnew transformation), used multiple datasets (Iris, Digits, Breast Cancer), fine-tuned Laplace smoothing, optimized visualization by improving scatter plot labelling and colouring, evaluated different scoring metrics (accuracy, precision, recall), analysed the impact of categorical vs. continuous features, explored the effect of alpha in smoothing for MultinomialNB | Deeper understanding of Naive Bayes variants, improved classification accuracy, impact of feature distributions, better interpretability of categorical vs. continuous data, ensured originality | 2.5 | 6 | Same notebook |