**Why is feature scaling important for gradient-based algorithms?**

* When features are on different scales, algorithms can struggle to converge in a timely manner due to features with larger values having greater influence. Feature scaling allows variables to not be normalized and more balanced. This improves accuracy and interpretability, in addition to a faster convergence time by preventing uneven weight updates and avoiding skewing the model due to bias toward larger features.

**Explain the difference between batch gradient descent and stochastic gradient**  
 **descent.**

* **Batch:**
* Uses the entire dataset to compute the gradient before updating parameters.
* Pros: Stable convergence, smooth loss curve.
* Cons: Quickly becomes computationally expensive with more data.
* Stochastic:
* Uses one random sample at a time to compute the gradient and update parameters.
* Pros: Much faster and allows training on larger datasets.
* Cons: Samples can be noisy, and the model requires more careful tuning of the learning rate.

**Why does scikit-learn Perceptron and Adline algorithms outperform book code. Research and develop an informed answer.**

* Scikit-learn adds regularization and optimization to construct perceptron and adaline models that are more efficient and robust than our book code. In detail, scikit-learn has:
  + Faster convergence: shuffle + adaptive learning rate + tol-based stopping.
  + Better generalization: built-in regularization.
  + Numerical stability: vectorized linear algebra vs Python loops.
  + Scalability: efficient on large, sparse datasets.
  + Ease of tuning: integrates with GridSearchCV to systematically optimize parameters.

* **Regularization Support** 
  + SGDClassifier (used for Adaline) applies **L2 regularization by default** (penalty='l2', alpha=0.0001), which helps prevent overfitting.
  + Perceptron does **not** use regularization unless explicitly enabled (penalty=None by default)
* **Optimized Learning and Convergence**
  + Both models use efficient optimization strategies like **stochastic gradient descent** with options for **early stopping**(early\_stopping=True) and **learning rate schedules** (learning\_rate='optimal' or 'adaptive')
* **Numerical Stability and Performance**
  + scikit-learn is built on **NumPy and Cython**, offering fast, stable, and vectorized computations that reduce rounding errors and improve training speed
* **Robust Defaults and Preprocessing Integration**
  + scikit-learn models are designed to work well with preprocessing tools like StandardScaler, and include defaults for hyperparameters that are tuned for general performance

**Compare the decision boundaries of logistic regression and SVM.**

* **Logistic Regression**
  + Shows a clear linear decision boundary.
  + Even when classes overlap, it fits a boundary based on predicted probabilities.
  + Always tries to separate classes as best as possible using a straight line.
* **Linear SVM**
  + No visible boundary, model predicts mostly one class.
  + If the two features don’t allow for linear separation, SVM may fail to find a meaningful dividing line.
  + This is likely why the graph shows only one dominant class region.
* **RBF SVM**
  + Displays a visible, curved decision boundary.
  + Uses Euclidean distance to measure similarity between points.
  + Can capture complex, non-linear patterns, better suited for overlapping like in this case

**What is the role of regularization in preventing overfitting?**

* Overfitting occurs when a model fits training data too closely and fails to generalize on new data. Regularization combats this by penalizing the number of features. By doing this, you limit how complex you can make the model, resulting in a model that is not overly tuned to the training data.
* **Overfitting** happens when a model learns the training data too well—including its noise and quirks—so it performs poorly on new, unseen data.
* **Regularization** helps by adding a penalty to the model’s complexity during training. This discourages the model from relying too heavily on any one feature or from fitting the data too precisely.
* Instead of just minimizing the error on the training set, regularization also tries to keep the model **simpler and more generalizable**.
* This is done by **shrinking the weights** of less important features, which reduces the risk of the model being overly tuned to the training data.
* The result is a model that may not fit the training data perfectly, but is more likely to perform well on new data.

**Vary the C values of the scikit-learn Logistic Regression and linear SVC models**  
 **with [0.01, 1.0, 100.0]. Discuss the impact.**

* Logistic Regression:
  + Results:
    - Accuracy was 0.83 for all three C values
    - Precision, recall, and F1 scores were very consistent
    - Confusion matrices very consistent
  + Why:
    - We used StandardScaler during preprocessing, which ensures all features are on the same scale. This makes regularization more balanced across features, so changing the C value doesn’t drastically shift the model’s behavior.
    - Our model also learned a stable decision boundary — meaning it consistently separates the classes well without needing extreme coefficient values or heavy regularization.
    - Finally, the class imbalance in our dataset had a stronger influence on performance than regularization did. Since the minority class is underrepresented, tuning C alone wasn’t enough to improve its recall or precision.
* SVM:
  + Results:
    - Accuracy was around .85 for all three C values with both the rbf and linear kernels
    - Adding PCA resulted in extremely similar results – also around 0.85
  + Why:
* Both models show consistent trends as C increases (less regularization):
  + Recall for the minority class (1) improves from ~0.57 to ~0.60–0.61
  + Precision for the minority class slightly decreases from ~0.72–0.73 to ~0.70–0.71
* Majority class (0) performance remains strong across all settings
* precision ~0.88–0.89, recall ~0.92–0.94
* Overall accuracy stays stable at ~85%, driven by class imbalance
* Trade-off: Increasing C allows the models to better detect minority class instances (higher recall), but at the cost of slightly more false positives (lower precision)
* Conclusion: Logistic Regression and Linear SVC behave similarly under varying regularization, with performance gains for the minority class as C increases