QUESTION 1:

The explanation of the Cost Function of Logistic Regression

Logistic regression is a widely used statistical method for binary classification problems, such as predicting whether a stock price will increase (1) or decrease (0). The cost function plays a crucial role in logistic regression as it measures the difference between the predicted values and the actual values. The cost function, also known as the loss function, quantifies the accuracy of the model's predictions. Specifically, the cost function for logistic regression is known as the log-loss or binary cross-entropy. The primary objective of this function is to minimize the discrepancy between the predicted probabilities and the actual classification outcomes.

Mathematical Expression

For a single training example, the cost function of logistic regression is defined as:

$$Cost(\hat{y}, y) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Where:

- $\hat{y}$ is the predicted probability that the outcome is 1.
- y is the actual outcome (0 or 1).

Properties of the Cost Function

1. Reward for Accurate Predictions:
   - When the actual outcome y=1 and the predicted probability $\hat{y}$ is close to 1, $\log(\hat{y})$ is close to 0, resulting in a low cost.
   - When the actual outcome y=0y = 0y=0 and the predicted probability $\hat{y}$ is close to 0, $\log(1 - \hat{y})$ is close to 0, resulting in a low cost.
2. Penalty for Incorrect Predictions:
   - When the actual outcome y=1 but the predicted probability $\hat{y}$ is close to 0, $\log(\hat{y})$ is a large negative value, resulting in a high cost.
   - When the actual outcome y=0 but the predicted probability $\hat{y}$ is close to 1, $\log(1 - \hat{y})$ is a large negative value, resulting in a high cost.

Overall Cost Function

For a dataset with m examples, the overall cost function J is the average cost across all examples:

$$J(w,b) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(\hat{y}^{(i)}) + (1-y^{(i)})\log(1-\hat{y}^{(i)})]$$

Here, $\hat{y}^{(i)} = \sigma(w^T x^{(i)} + b)$, where σ is the sigmoid function, mapping the linear combination to a value between 0 and 1.

Optimization Process

To optimize the model, gradient descent is commonly used to minimize the cost function. This involves calculating the derivatives of the cost function with respect to the model parameters and updating the parameters iteratively to reduce the cost:

$$w := w - \alpha\frac{\partial J}{\partial w}$$

$$b := b - \alpha\frac{\partial J}{\partial b}$$

Where α is the learning rate, controlling the step size of each update.

Conclusion

The cost function in logistic regression measures the error between predicted probabilities and actual outcomes. By minimizing the cost function, we can determine the optimal model parameters, thereby improving prediction accuracy. This is particularly important in financial forecasting, where accurate predictions can lead to significant economic gains.

QUESTION 2:

The explanation of Voting classifiers in ensemble learning

Voting classifiers are a type of ensemble learning method used in machine learning to improve the predictive performance of a model by combining the predictions of multiple base models (often called "weak learners"). The idea is to leverage the strengths of different models to produce a more accurate and robust overall prediction. There are two main types of voting classifiers: hard voting and soft voting.

Types of Voting Classifiers

1. **Hard Voting Classifier**:
   - **Definition**: In a hard voting classifier, each base model makes a prediction (votes for a class), and the final prediction is made based on the majority class that receives the most votes.
   - **Mechanism**: Suppose there are three base models (Model A, Model B, and Model C) and we are trying to classify a sample as either Class 0 or Class 1. If Model A and Model B predict Class 1, and Model C predicts Class 0, the hard voting classifier will predict Class 1, as it receives the majority vote.
   - **Formula**:

$$\hat{y} = mode\,(\hat{y}_1, \hat{y}_2, \ldots\ldots, \hat{y}_n)$$

   where $\hat{y}_i$ is the prediction of the i-th base model.

2. **Soft Voting Classifier**:
   - **Definition**: In a soft voting classifier, each base model provides a probability (or confidence level) for each class, and the final prediction is made based on the average (or sum) of these probabilities. The class with the highest average probability is chosen as the final prediction.
   - **Mechanism**: If the same three models (Model A, Model B, and Model C) provide probabilities for Class 0 and Class 1, the soft voting classifier will average these probabilities and choose the class with the highest average probability.
   - **Formula**:

$$\hat{y} = arg\max_{c}\left(\frac{1}{n}\sum_{i=1}^{n} P\,(y = c \mid X,\ Model_i)\right)$$

where $P\left(y = c \mid X,\ Model_i\right)$ is the probability of class c given the input X and the i-th base model.

Advantages of Voting Classifiers

1. **Improved Accuracy**: By combining multiple models, voting classifiers often achieve higher accuracy compared to individual models.
2. **Robustness**: They can reduce the risk of overfitting, as the errors of individual models can be offset by the correct predictions of others.
3. **Simplicity**: Voting classifiers are relatively simple to implement and interpret compared to more complex ensemble methods like boosting or stacking.

Disadvantages of Voting Classifiers

1. **Computational Cost**: Combining multiple models can be computationally expensive, especially if the base models are complex.
2. **Diminishing Returns**: Adding more models does not always lead to better performance, especially if the models are not diverse.

QUESTION 3:

Please see the python notebook, the code and related details are given in that document.