CODE BASICS

MACHINE LEARNING TAKEAWAYS

Chapter:

Supervised Machine Learning: Classification

Introduction to Classification

1. Regression Models work with continuous values that can take on any value, while Classification is categorical and sticks to a definite set of values.

2. The Classification Model can be split into two types:
    - Binary Classification
    - Multiclass Classification

Logistic Regression: Binary Classification

1. The Sigmoid Function converts input into a range from 0 to 1.

   sigmoid (z) = 1 / (1+ e^(−z))

   Where e is Euler's number = 2.71828

2. It is a crucial function in logistic regression for binary classification as it maps linear outputs to probabilities, helping to improve the model's predictions.

Model Evaluation: Accuracy, Precision, Recall

1. Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined.

   Accuracy = (True Positives + True Negatives) / Total Cases

2. While accuracy can be a useful metric, it might not always provide a comprehensive view of a model's performance. Therefore, we also consider other measures like precision and recall to ensure a more rounded evaluation.

Model Evaluation: Accuracy, Precision, Recall

3 Precision is the ratio of true positive predictions to the total predicted positives, measuring the accuracy of the positive predictions.

Precision = True Positives/ (True Positives + False Positives)

4 Recall is the ratio of true positive predictions to the actual positives, assessing the model's ability to identify all relevant cases.

Recall = True Positives/ (True Positives + False Negatives)

Model Evaluation: F1 Score, Confusion Matrix

1. The F1 score is the harmonic mean of the precision and recall.

   F1 Score = 2 x (Precision + Recall) / (Precision x Recall)

2. A confusion matrix is a table layout that visualizes the performance of a classification algorithm by displaying the true and false predictions it makes.

Logistic Regression: Multiclass Classification

1. The sklearn library contains a range of sample datasets that are excellent for learning.

2. To enhance understanding, it's beneficial to include relevant images and documentation alongside your code in the Jupyter notebook.

Cost Function: Log Loss

1. The Cost Function (MSE) will work for Linear Regressions as it is a convex function, but it won't work for Logistic Regression as it is a non-convex function.

2. Log Loss Formula –

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{n}\left[y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)\right]$$

- y_i = Actual value for the i-th record
- p_i = Predicted probability for the i-th record
- N = Number of Records

Cost Function: Log Loss

3 Log Loss is the underlying Cost Function that we use in Logistic Regression.

4 Log Loss is also known as Logistic Loss or Binary Cross Entropy, or Multinomial Log Loss.

Support Vector Machine (SVM)

1. Support Vector Machine (SVM) is a robust supervised learning model that finds the optimal hyperplane in an n-dimensional space for classification and regression tasks.

2. A Kernel is a function that transforms data into a higher dimensional space so that a decision boundary can be drawn. There are different kernels available for different use cases, some examples - poly, linear, rbf, sigmoid.

3. **Gamma (γ):** This is a parameter in SVM that decides the impact of each data point on the decision boundary. It's about how closely the model adheres to the data.

Support Vector Machine (SVM)

4 **Regularization:** In machine learning, regularization adds a complexity penalty to the model to curb overfitting and enhance its performance on new data.

5 Choosing the appropriate kernel significantly influences the total computational power. This, in turn, affects the budget of the entire project.

Data Preprocessing: Scaling

1. **S**caling in machine learning involves adjusting the range of feature values to a common scale, such as 0 to 1, enhancing model performance by ensuring each feature contributes equally.

    - Min-Max Scaling
    - Standard Scaling (Z-score normalization)

Sklearn Pipeline

1. Sklearn offers a handy and robust tool known as pipeline. This feature lets you build and operate a series of data transformation and modeling tasks as one entity.

2. This tool is instrumental in simplifying machine learning workflows, safeguarding uniformity in data processing, and enhancing the efficacy of model creation and evaluation.

Naive Bayes: Theory

1. P(A | B) = Probability of event A knowing that event B has already occurred.

2. Bayes' Theorem ->

   $P(A|B) = P(B|A) * P(A) / P(B)$

3. It's called Naive Bayes because it makes the naive assumption that all features (such as p(free) or p(lottery)) are independent of each other.

Naive Bayes: SMS Spam Classification

1. Machine learning models process numbers, not text. Hence, during preprocessing, it's crucial to convert text-based datasets into numerical form for model training.

2. The Count Vectorizer transforms a collection of text documents into a matrix of token counts. It essentially reflects the frequency of each word within the documents. Sklearn offers a built-in API for this operation.

3. For the Naive Bayes implementation, we opted for MultinomialNB. It's apt for situations where features are word frequencies in a text, making it a popular choice for text classification.

Decision Tree: Theory

1 In machine learning, a decision tree is a type of supervised learning algorithm. It models decisions and their potential outcomes in a structure resembling a tree, composed of various choices.

2 You can select the higher-level notes using either of the following typical methods.

- **Gini Impurity**

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$

- **Entropy (Information Gain)**

$$Entropy = -\sum_{i=1\ pi}^{k} \log_2(p_i)$$

pi = Probability of item being classified into class i
k = Total number of classes

Handle Class Imbalance: Theory

1. Class imbalance in machine learning occurs when the number of samples in each class is not equal, leading to a skewed distribution. There are several techniques to handle class imbalance in machine learning:

    - Under Sampling Majority Class: Reduce instances in the majority class.

    - Over Sampling minority class by duplication: Create more minority class instances by duplicating them.
        - Over sampling minority class using SMOTE
            - Generate synthetic examples using k nearest neighbors algo.

Handle Class Imbalance: Theory

- SMOTE – Synthetic Minority Over-sampling Technique

  - Over sampling – SMOTE Tomek Links: Remove bridges (Tomek links) between neighboring minority and majority class instances.

2. Ensemble Method: Combine models to balance class imbalance.