4/22/2024

Classification

Grouping the data into distinct classes or categories (categorical or discrete).
Binary Classifier, Multi-class classifiers, Multi-label classifiers

Lazy learner vs. eager learner

Lazy learner predicts directly from the training data itself
Eager learner will create a model, and then base predictions on that model (such as regression)

Logistic regression – binary classifier

Naïve Bayes classifier: needs very little data. Fast. But unfortunately it's a poor predictor. (based on a two-way table/crosstabs, and assumes independence.)

Stochastic gradient descent – many hyperparameters, it can be sensitive to scaling, depends on the derivative to find the best "descent" path to find the minimum error.

KNN – K nearest neighbors – finds the shortest distance to a value (or k values) in the training set, and then assigns the test value the same category (voting is applied, majority wins). It's best to use odd number of nearest neighbors, because ties are randomly decided.

Decision Trees – can be quite unstable. It is generally overfitted. (test data never does as well as training data) – bagging

Random Forest – ensemble method; builds a number of small trees, using random combinations of variables, and then "votes" on which class the predicted value belongs to. – boosting

Neural Network – high noise tolerance, hard to interpret

SVM – support vector machines – memory efficient, useful for high dimensional data. The simplest form uses a line to divide the categories (a plane, hyperplane). The plane/line is in the middle of all the data. That line is placed so that the mistakes are as few as possible. The optimal separation line. Can project onto higher dimensions to obtain non-linear separators. (also a binary classifier)

Evaluate classifiers:
Test/train split – hold-out method
Cross validation – k-fold cross validation
Accuracy – measured as a percent of correctly classified values
F-1 score (weighted average of precision and recall)
ROC curve

General methods:

- Read the data
- Create dependent and independent data sets
- Split into test/train
- Train the model with different classifiers

- Choose the best classifiers with the most accuracy

Regression

Linear Regression – ordinary least squares

Nonlinear Regression – polynomials models, splines, log models, power models, etc. and can include interaction terms.

Model Selection:

Stepwise method
Best Subset selection
PCA – principal component analysis
PLA – principal least squares
Penalized Regression – LASSO, Ridge regression, etc.

Metrics:

Root mean square error (RMSE)
Adjusted R^2
K-fold cross validation

Graph Theory:

Deep learning – neural networks
Tree-based models
Markov chain models

Python –

Sklearn (sci-kit learn), contains most major machine learning tools
Keras tensor flows, pytorch for neural networks

Data Equity

Especially when dealing with classification models, where one group or class is significantly smaller than the other classes. This is called masking. It may be more efficient (accuracy) to overlook the smaller class and mispredict that class in order to get higher accuracy on the more dominant class. You may get high overall accuracy, but the small class may be consistently mispredicted.

This is problematic, especially when dealing with people and groups of people that are minorities, or are systematically underrepresented. The models can then reproduce the biases of the culture that the data was drawn from.

When the classes are of unequal sizes, you may need to adjust the class sizes in the training data to produce more equally sized input groups. Spread the inaccuracy over all the groups instead of just the one.

Sentiment Analysis:

A type of NLP, to detect positive or negative sentiment, other feelings like anger, or urgency, intention or interest

Mutli-lingual sentiment analysis – not very good at switching between languages. Now use language detection tool to identify the language, then use a single-language sentiment analysis method.

Resources:

1. https://www.geeksforgeeks.org/clustering-in-machine-learning/
2. https://www.edureka.co/blog/classification-in-machine-learning/
3. http://www.sthda.com/english/wiki/regression-analysis-essentials-for-machine-learning
4. https://towardsdatascience.com/graph-theory-and-deep-learning-know-hows-6556b0e9891b
5. https://machinelearningmastery.com/statistical-hypothesis-tests/
6. https://monkeylearn.com/sentiment-analysis/
7. https://www.abtassociates.com/insights/podcast/data-racial-equity-how-do-we-eliminate-bias-from-ai-machine-learning-and-more

Extended commentary:

Classification is a type of supervised learning method in which a model is trained to predict the class or category of a given input data based on a set of labeled examples. Classification methods are widely used in various fields such as image recognition, speech recognition, sentiment analysis, fraud detection, and many others. Some popular classification algorithms include:

Logistic Regression: This algorithm models the probability of a given input belonging to a particular class using a logistic function.

Naive Bayes: This algorithm is based on Bayes' theorem and assumes that all features in the input data are independent of each other.

Decision Trees: This algorithm builds a tree-like model of decisions and their possible consequences.

Support Vector Machines (SVM): This algorithm constructs a hyperplane or set of hyperplanes that separates the different classes in the input data.

K-Nearest Neighbors (KNN): This algorithm classifies new inputs by finding the k-nearest neighbors in the training data and using the majority class among those neighbors.

Random Forest: This algorithm creates an ensemble of decision trees and uses their collective output to make a classification.

Neural Networks: This algorithm is based on a network of interconnected nodes or neurons that can learn to recognize patterns in the input data and make a classification.

Each of these algorithms has its strengths and weaknesses and is suited to different types of input data and classification tasks.

In machine learning, there are two main types of learning approaches: eager learning and lazy learning.

Eager learners, also known as model-based methods, construct a classification model using the training data before receiving test data. The model is then used to make predictions on new data. Eager learners require a significant amount of time and computational resources during training, but are faster during the prediction phase.

On the other hand, lazy learners, also known as instance-based methods, delay the construction of a classification model until the test data is presented. Rather than constructing a model, the lazy learner stores the training data and waits for a new instance to be classified. When a new instance is presented, the learner searches the training data for similar instances and makes a prediction based on the class labels of those instances. Lazy learners require less time during the training phase, but are

slower during the prediction phase as they have to search through the entire training set for each new instance.

In summary, the main difference between eager and lazy learners is that eager learners construct a model during the training phase, while lazy learners store the training data and search it during the prediction phase.

Classification models are evaluated using various performance metrics, which assess the accuracy and effectiveness of the model in predicting the class label of new data instances. Some of the commonly used evaluation metrics for classification models include:

Confusion Matrix: A matrix that summarizes the number of true positive, true negative, false positive, and false negative predictions made by the model.

Accuracy: The proportion of correct predictions made by the model out of the total number of predictions.

Precision: The proportion of true positive predictions out of the total positive predictions made by the model.

Recall: The proportion of true positive predictions out of the actual number of positive instances in the dataset.

F1-score: The harmonic mean of precision and recall, which provides a balanced measure of both metrics.

ROC curve: A graphical representation of the performance of the model that shows the trade-off between the true positive rate and false positive rate at different classification thresholds.

AUC: The area under the ROC curve, which provides a single metric to evaluate the overall performance of the model. A higher AUC indicates a better-performing model.

The choice of evaluation metric depends on the problem domain and the specific requirements of the application. For example, in a medical diagnosis application, recall might be more important than precision, as it is more critical to correctly identify all positive cases, even if some false positives are included.

Data equity refers to the fair and ethical use of data in data analysis. It recognizes that data can impact individuals, groups, and society as a whole and aims to ensure that data is used in a just and equitable manner.

In practice, data equity means that data analysis should be conducted in a way that is fair, unbiased, and transparent. This includes considering how data was collected and whether it may contain any implicit biases or representational gaps. It also means being aware of the potential impact that data analysis may have on different groups of people and taking steps to mitigate any negative consequences.

To ensure data equity, it is important to have diverse perspectives represented in the data analysis process. This can include having people from different backgrounds and experiences involved in collecting and analyzing data, as well as considering the perspectives and needs of different stakeholders in the interpretation and use of data. Additionally, it may involve considering ethical guidelines or legal regulations that govern the use of data, such as data privacy laws or regulations that prohibit discrimination.

Sentiment analysis is a natural language processing technique used to determine the sentiment or emotion conveyed by a piece of text. It involves the use of algorithms to analyze text and identify the sentiment expressed in the text. The sentiment can be positive, negative, or neutral.

The process of sentiment analysis involves several steps:

Text preprocessing: The first step in sentiment analysis is to preprocess the text to remove any noise or irrelevant information from the text. This involves tasks such as tokenization, stop word removal, and stemming.

Feature extraction: The next step is to extract features from the text that can be used to determine the sentiment. This involves identifying important words or phrases in the text that can be used to determine the sentiment.

Sentiment classification: Once the features have been extracted, the sentiment of the text can be classified using various classification algorithms such as Naive Bayes, Support Vector Machines (SVM), or neural networks. These algorithms assign a sentiment score to the text based on the features extracted in the previous step.

Evaluation: The final step is to evaluate the performance of the sentiment analysis model. This is done by comparing the predicted sentiment scores with the actual sentiment of the text. Common evaluation metrics include accuracy, precision, recall, and F1-score.

Sentiment analysis can be used in a variety of applications, such as social media monitoring, brand reputation management, and customer feedback analysis.