recap

05 February 2024 13:5

- Juypter note books and pycharm

Neo4j

Machine learning

Api

Network x

- Numpys, pandas, matplotlib, seaborn, plotly

- Api development flask, fast api

- Regular expressions
- TF-IDF such as word2vec
- ML techniques
- Network x
- Neo4j
- Graph alogorithim centrality measure

- Graph data modeling

Machine learning

Supervised learning

Unsupervised learning

05 February 2024

14:2

-

Supervised learning error metrics

Supervised learning algorithms are trained labelled examples, such as input where desired output is known.

-

Unsupervised learning error metrics

Unsupervised learning algorithms find structu labelled, classified or categorized

-

-

Either correct or

- Commonly used for:

- Known inputs and outputs to train the algorithm, uses this to make predictions on new unseen data
- Commonly used where historic data predicts future events

  - Clustering

  - Dimensionality reduction

- Data split 3 sets:
  - Training data, Validation data, test data

- Two types of problems:
    - Regression: mapping predictive relationship between labels and data points

- <u>Classification:</u> Predict correct label for input data

- Regression: linear regression

- Classification: K-NN, SVM, random forest, decision trees, naïve bayes, logistic regression

Supervised learning

12 February 2024 12:2

Classification error metrics

Regression error metrics

- Mean absolute error, mean of absolute value of er

- With classification the can either be correct or incorrect

- Large errors not really punished

- **Accuracy**, number of correct prediction divide by overall predictions

- Mean square error, mean of squared error

- Large error punished more

- Root mean square error, root of MSE

- Pros: useful when classes are balanced

- Punishes large errors again

- Cons: not useful when classes are unbalanced

- **Recall:** to find all relevant cases within dataset, true positives divided by true positives + false negatives

- **Precision:** ability to identify only relevant data points, true positives divided by true positives + false positives

- Will have trade-off between recall and precision

- F1 score takes harmonic mean of precision vs recall

models

Linear regression:

Decision trees random forest:

05 February 2024 14:02

Logistic regression:

Support vector machines:

- Predictive modelling: used when linear relationship between input and output values i.e square footage and house price

KNN:

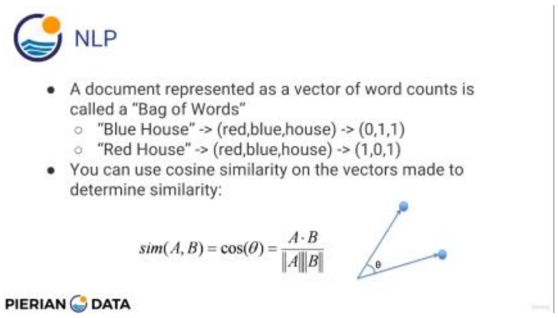
- Tree method:
- Calculates distance x from all points in your data
- Allow us to solve classification problems where we are trying to predict discrete values
- Supervised learning models that analyse learning models with associated learning algorithms to analyse data and recognise patterns.
   - Root: tree that performs a split
   - Leaves: terminal node that predict outcome
- Sorts points by increasing distance from x
- Predict majority label of "k" closest points
- Relationships: between independent and dependent variable, can analyse coefficients
- Use confusion matrix to analyse it
- Used for regression and classification
- Random forests: to improve performance we can use many trees with random sample of features chosen as split
- Each data point is assigned a class based on the class labels of its neighbours

- Used mainly for **binary classification** problems
- Aims to find hyper plane between data points that best seperates data points in different classes, while maximising margin. Which is distance between hyper plane and nearest data point (support vector) (classification)
- Feature selection: work out with features to see which have most impact on outcome variable
- Uses sigmoid function to transform to range from [0,1], then to classify
- If there's one very strong feature in data set, when using bagged trees most trees will have this feature as top split. Resulting in ensemble of trees that are highly correlated
- (add more)
- **Pros**: very simple, training is trivial, works with any number of classes, easy to add more data, few parameters (distance metric, k)
- SVMs can handle non-linear decision boundaries by mapping the input features into a higher-dimensional space using kernel functions.
- Extrapolation and interpolation: can be used to work out these two
- Can analyse features to see coefficients which indicate relation to target class.
- **Cons**: high prediction cost, not good high dimensional data, categorical features don't work well

- Can be regularised to prevent overfitting

nlp

19 February 2024 09:03

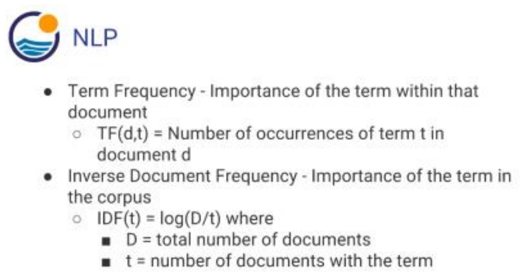


NLP

- A document represented as a vector of word counts is called a "Bag of Words"
  - "Blue House" -> (red,blue,house) -> (0,1,1)
  - "Red House" -> (red,blue,house) -> (1,0,1)
- You can use cosine similarity on the vectors made to determine similarity:

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

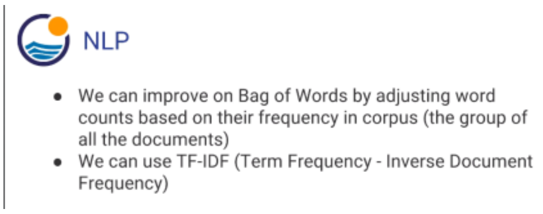
PIERIAN DATA



NLP

- Term Frequency - Importance of the term within that document
  - TF(d,t) = Number of occurrences of term t in document d
- Inverse Document Frequency - Importance of the term in the corpus
  - IDF(t) = log(D/t) where
    - D = total number of documents
    - t = number of documents with the term

Screen clipping taken: 19/02/2024 11:48



NLP

- We can improve on Bag of Words by adjusting word counts based on their frequency in corpus (the group of all the documents)
- We can use TF-IDF (Term Frequency - Inverse Document Frequency)

Vectorization:

- Convert each message represented as a list of tokens (lemmas), into a vector that , machine learning models can understand
- Do that in 3 steps with bag of words model:
  - Count how many times does a word occur in each message (term frequency)
  - Weight the counts, so that frequent tokens get lower weight (inverse document frequency)
  - Normalize vectors to unit length, to abstract from original text length

Use pipeline?

Deep learning

19 February 2024 11:54

Steps:

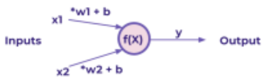
1. Single biological neuron
2. Perceptron
3. Multi-layer perceptron model
4. Deep learning neural network

Other concepts:

1. Activation functions
2. Gradient descent
3. Back propagation

Perceptron model:

- y = (x1w1 + b) + (x2w2 + b)



- We've been able to model a biological neuron as a simple perceptron! Mathematically our generalization was:

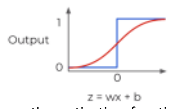
$$\hat{y} = \sum_{i=1}^{n} x_i w_i + b_i$$

Neural network:

- Can expand on idea of single perceptron, to create a multi-layer perceptron model
- Outputs of one perceptron act as inputs to another perceptron
- Hidden layers: layers in-between input and output layers
- Deep neural networks: contain 2 or more hidden layers

Activation functions:

- Z = x*w + b
- Pass z through activation function to limit its value
  - Lucky for us, this is the sigmoid function!



$$f(z) = \frac{1}{1 + e^{(-z)}}$$

- Wiki to see other activation functions

Multi class classification problems:

2 types of situations:

- Non - exclusive classes
  - Data point to which multiple classes/categories assigned to it
- Mutually exclusive classes
  - Only one class per data point
  - e.g photo can either be colour or grayscale

Organizing multiple classes:

- Have 1 output node per class
- Use one hot encoding (dummy variables)
- After data organised correctly, choose correct classification function for output layer
- Non-exclusive: sigmoid function
- Mutually exclusive: softmax function

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K$$

-

Cost functions:

- Output y models estimation of what it predicts the label to b. How do we evaluate it?
- Need to take estimated outputs and compare with real values of the label
- Cost function is a average
- Calculate difference between actual and predicted values

$$C = \frac{1}{2n} \sum_{x} \| y(x) - a^L(x) \|^2$$

- Want to minimise cost function, what value of w results in minimum of c(w)
- Use gradient descent to solve problem
- Step size is learning rate
- For classification problems use *cross entropy loss function*

Backpropagation:

- Want to know how cost function changes with respect to weights in the network, so can update weights to minimize cost func
- How sensitive cost function to changes in w

$$\frac{\partial C_0}{\partial w^L} = \frac{\partial z^L}{\partial w^L} \frac{\partial a^L}{\partial z^L} \frac{\partial C_0}{\partial a^L}$$

- Lots of maths steps?

models

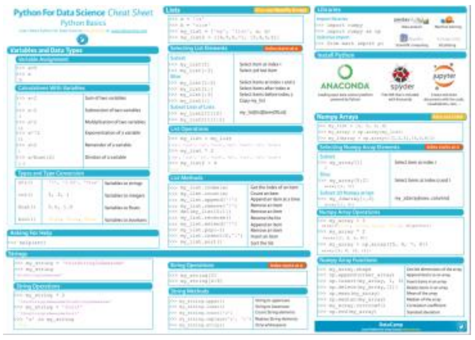
12 February 2024 15:08

<u>K means clustering:</u>

- Unsupervised algorithm which will attempt to group similar clusters together in your data
- Typical clustering problems:
  - Clustering similar documents
  - Cluster customers based on features
  - Identify similar physical groups
- Divide data into distinct groups
- Choose number of clusters "k", randomly assign each point to cluster. Keep repeating until clusters stop changing

Cheat sheets

12 February 2024 12:58



Python For Data Science Cheat Sheet
Python Basics

Variables and Data Types

Lists

Libraries

Install Python

Numpy Arrays

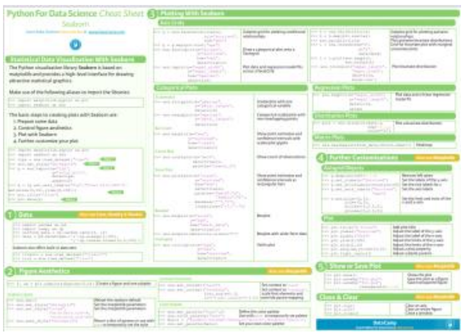
Asking For Help

Strings

String Operations

String Methods

DataCamp



Python For Data Science Cheat Sheet
Seaborn

Statistical Data Visualization With Seaborn

1 Data

2 Figure Aesthetics

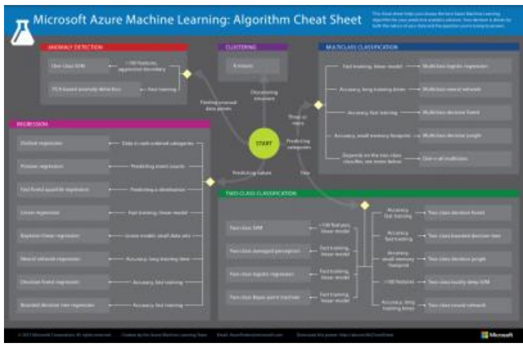
3 Plotting With Seaborn

4 Further Customizations

5 Show or Save Plot

Close & Clear

DataCamp



Microsoft Azure Machine Learning: Algorithm Cheat Sheet

ANOMALY DETECTION

CLUSTERING

MULTICLASS CLASSIFICATION

REGRESSION

START

TWO-CLASS CLASSIFICATION

Screen clipping taken: 12/02/2024 13:08

Python For Data Science Cheat Sheet
Keras

Keras

Data

Preprocessing

Model Architecture

Inspect Model

Compile Model

Model Training

Evaluate Your Model's Performance

Prediction

Save/ Reload Models

Model Fine-tuning

DataCamp

Python For Data Science Cheat Sheet
Scikit-Learn

Scikit-learn

Loading The Data

Training And Test Data

Preprocessing The Data

Create Your Model

Model Fitting

Prediction

Evaluate Your Model's Performance

Tune Your Model

DataCamp

Screen clipping taken: 12/02/2024 13:06

Screen clipping taken: 12/02/2024 13:05

OOP

19 February 2024 13:17