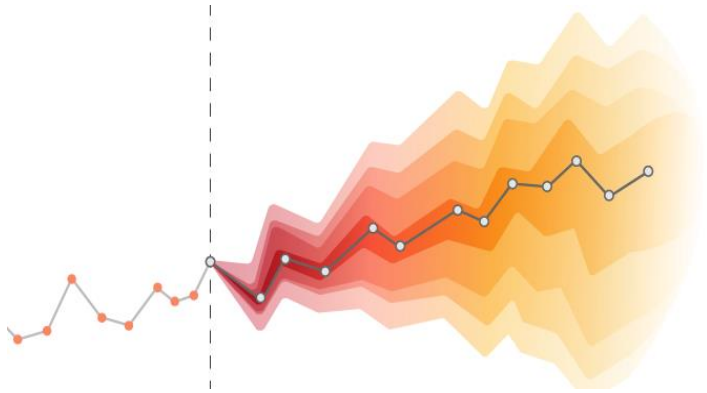
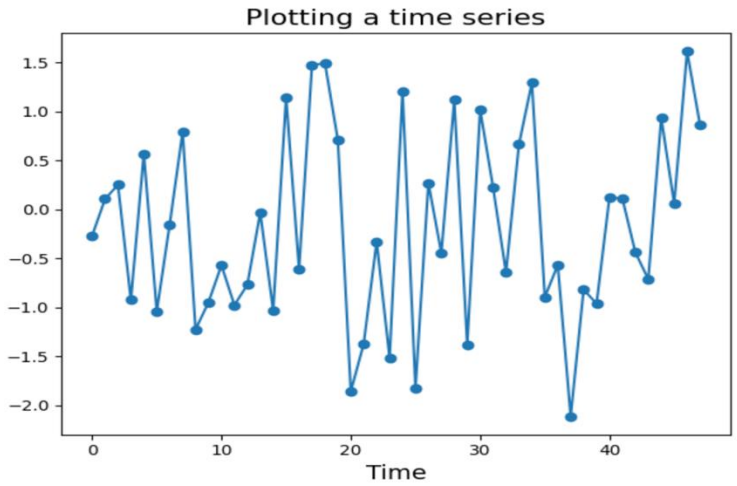
SAX-VSM Timeseries Data Classification

Muhammad Adeel Sultan Khan





Agenda

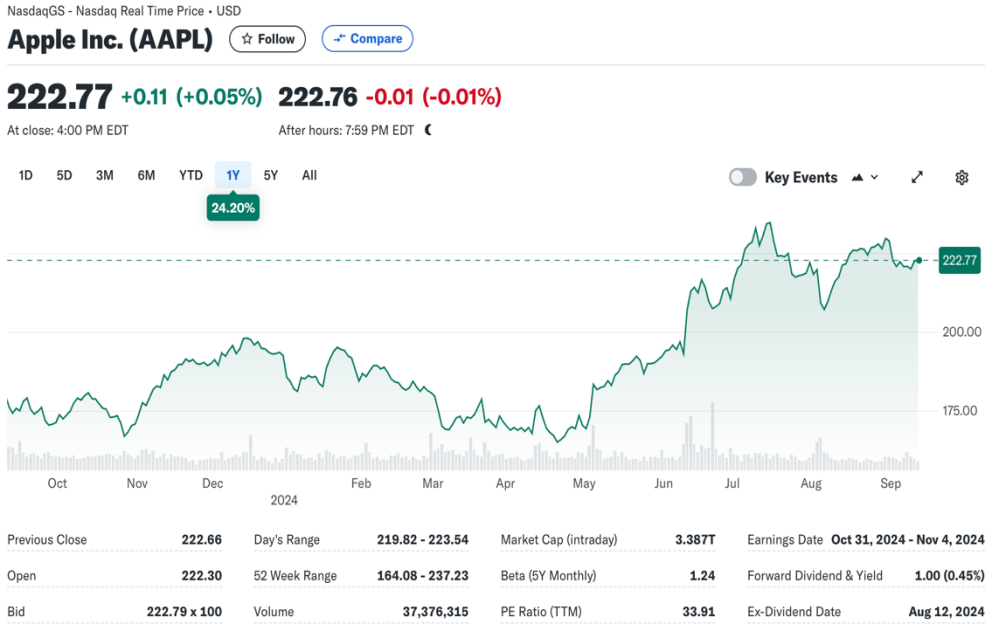
- What is timeseries data
- Timeseries data analysis
- Timeseries data analysis types
- Characteristics of timeseries data
- Timeseries data classification
- Shapelet transformation
- Bag of words for timeseries
- Symbolic Aggregate Approximation-Vector Space Modeling (SAX-VSM) algorithm
- Piecewise aggregate approximation (PAA)
- Dimensionality reduction vs Numerosity reduction
- Vector space modelling (tf-idf)
- Cosine Similarity

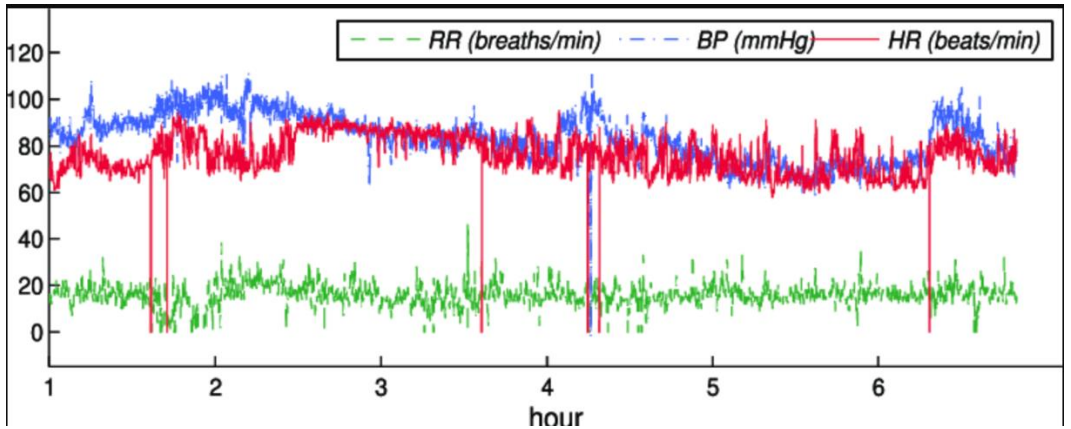
What is Timeseries Data

- Time series data is a sequence of data points collected at regular intervals over a period of time, where the order of the data points is crucial for understanding trends, patterns, and any variations that may exist within the data. Also known as time-stamped data

- Time series data is data collected at different points in time, such as GDP of a country by year, stock price of a company over a period of time, or heartbeat recorded at each second

- Time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.

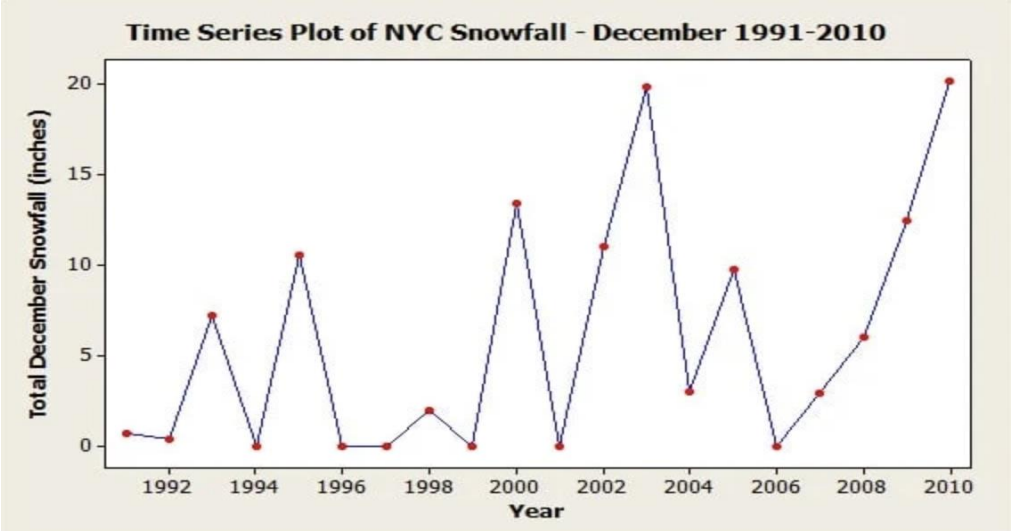


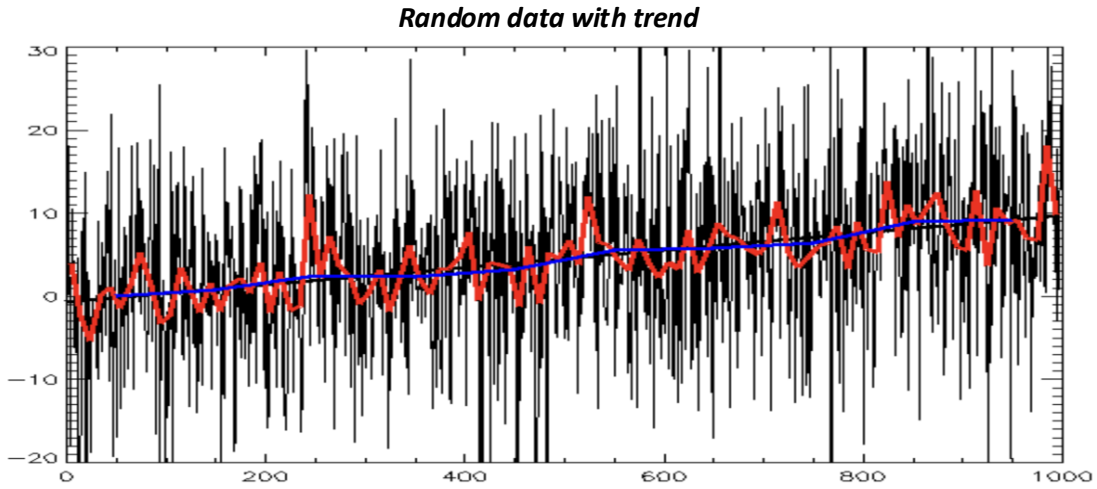


Timeseries Data Analysis

- Time series analysis is used for non-stationary data changing over time. Industries like finance, retail, and economics use time series analysis as currency and sales are always changing. Time series analysis is ideal for forecasting weather changes, helping meteorologists predict weather report and future climate changes. Examples of time series analysis include:

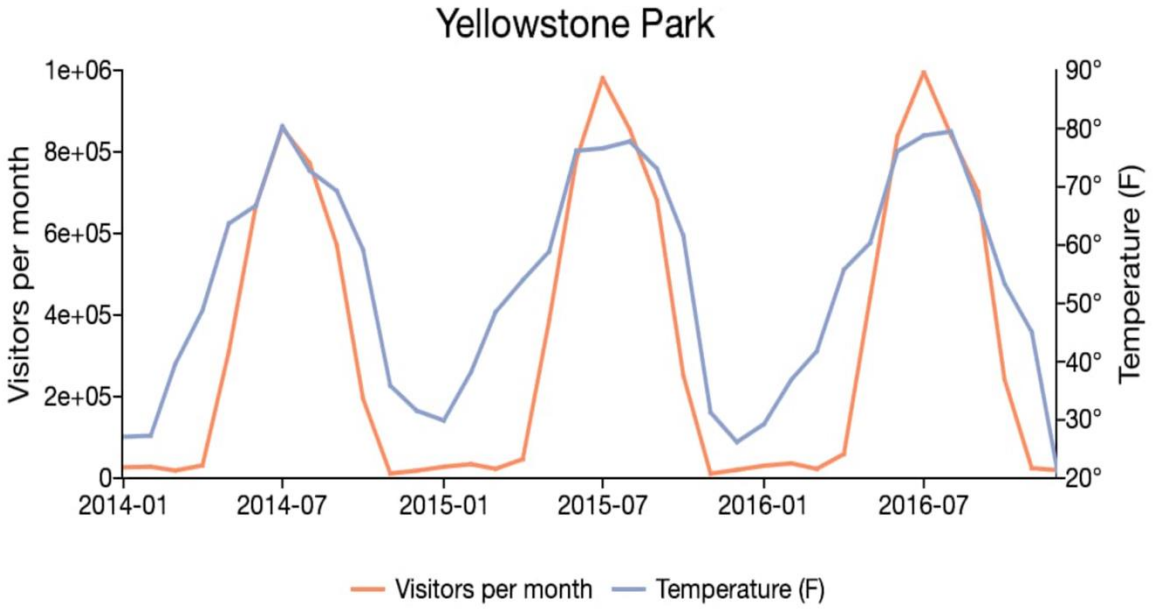
  - ❖ Weather data
  - ❖ Rainfall measurements
  - ❖ Temperature readings
  - ❖ Heart rate monitoring (EKG)
  - ❖ Brain monitoring (EEG)
  - ❖ Quarterly sales
  - ❖ Stock prices
  - ❖ Automated stock trading
  - ❖ Industry forecasts
  - ❖ Interest rates





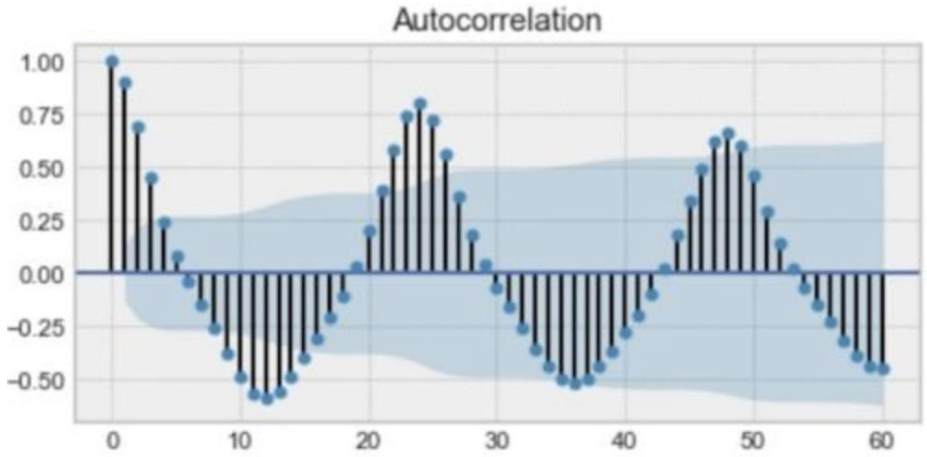
Timeseries Data Analysis Types

- **Classification:** Identifies and assigns categories to data.
- **Curve fitting:** Plots data along curve to study relationships of variables.
- **Descriptive analysis:** Identifies patterns in time series data, like trends, cycles, or seasonal variation.
- **Explanative analysis:** Attempts to understand data and relationships within it, as well as cause and effect.
- **Exploratory analysis:** Highlights characteristics of the time series data
- **Forecasting:** Predicts future data. This is based on historical trends. It uses the historical data as a model for future data, predicting scenarios that could happen along future plot points.
- **Intervention analysis:** Studies how an event can change the data.
- **Segmentation:** Splits the data into segments to show the underlying properties of the source information.

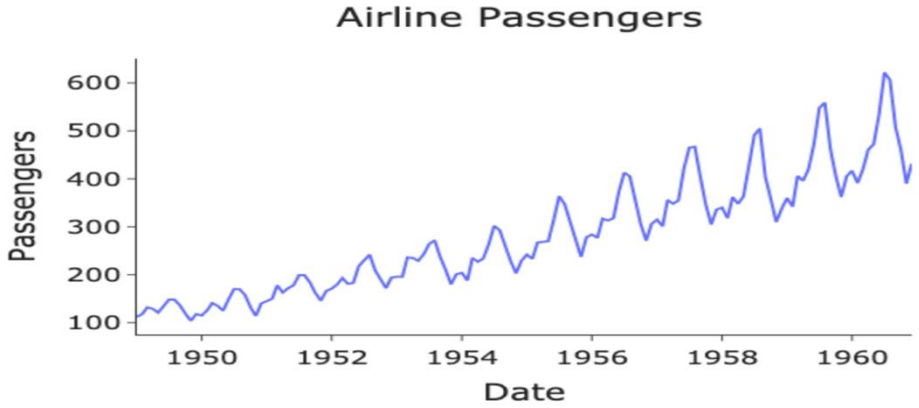


Characteristics of Timeseries Data

- ***Autocorrelation*** is the similarity between observations as a function of the time lag between them. An autocorrelation of +1 represents a perfect positive correlation, while an autocorrelation of -1 represents a perfect negative correlation. It can also be referred to as lagged correlation or serial correlation, as it measures the relationship between a variable's current value and its past values

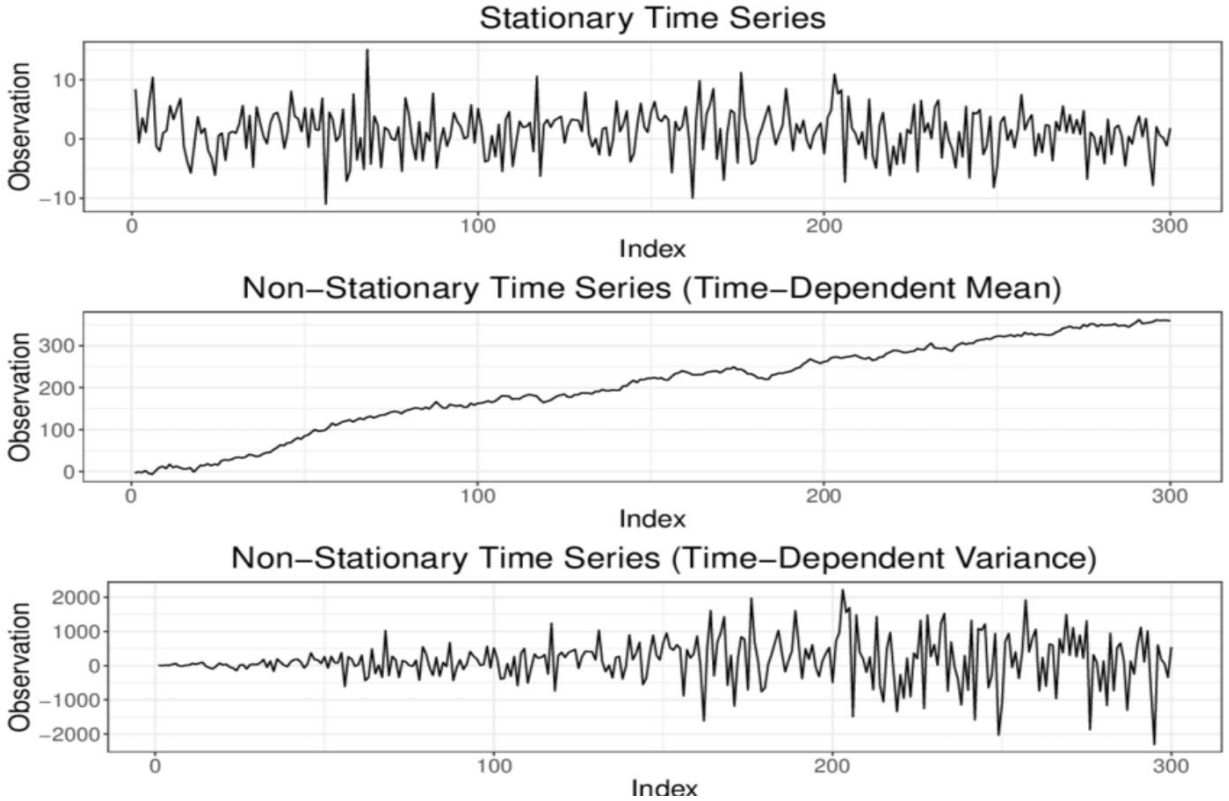


- ***Seasonality*** refers to periodic fluctuations in business areas and cycles that occur regularly based on a particular season. A season may refer to a calendar season such as summer or winter, or it may refer to a commercial season such as the holiday season



Characteristics of Timeseries Data

- ***Stationarity*** is a time series' status when its statistical properties, such as its mean, variance, and covariance, remain constant over time. Time series with trends or seasonality are not stationary because these factors affect the value of the time series at different times. To determine if a time series is stationary, you can look for specific characteristics in the data or use statistical tests. The Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test are two common statistical tests for this purpose



https://www.researchgate.net/figure/Examples-for-stationary-and-non-stationary-time-series_fig3_348592737

Timeseries Data Classification

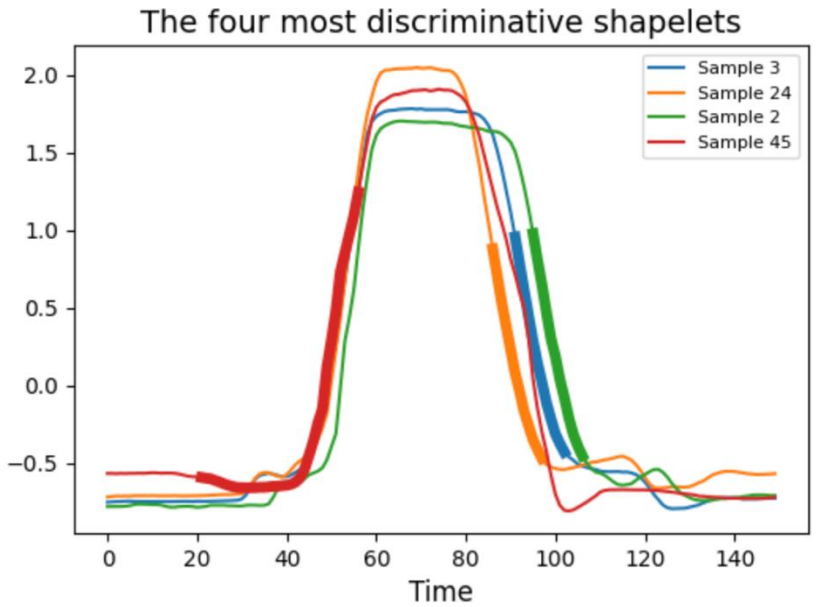
- Time series classification uses supervised machine learning to analyze multiple labeled classes of time series data and then predict or classify the class that a test data set belongs to. This is important in many environments where the analysis of sensor data or financial data might need to be analyzed to support a business decision

- ***Types of time series classification:***

  - ***Distance-based approaches:*** Euclidean, Hamming, Manhattan, Minkowski, K-nearest neighbors(KNN), Dynamic Time Warping (DTW)
  - ***Shapelet:*** Identifying data shapes in time series subsequences e.g ECG shapes to identify heart disease
  - ***Model ensembles:*** Collection of classification models that each perform their own class discrimination on the data set
  - ***Dictionary approaches:*** No. of occurrences of a particular shapelet in time series. Bag-of-Patterns (BoP) algorithm that looks at amplitude of a time series signal within some specified window of the data and applies a simple transformation to the signal to represent the mean of the signal within the window. This feature then becomes the basis for the dictionary that is used to train a classifier.
  - ***Interval-based approaches:*** Splitting time series into distinct intervals. Each interval is used to train an individual machine learning model (classifier)
  - ***Deep learning:*** Type of neural network that has multiple layers of neurons or perceptrons. These models are more complex with many more parameters than other types of algorithmic models. The initial layers of the deep learning network encode shapes within time series data, while latter layers encode representations that can be discriminated into classes in the final layer.

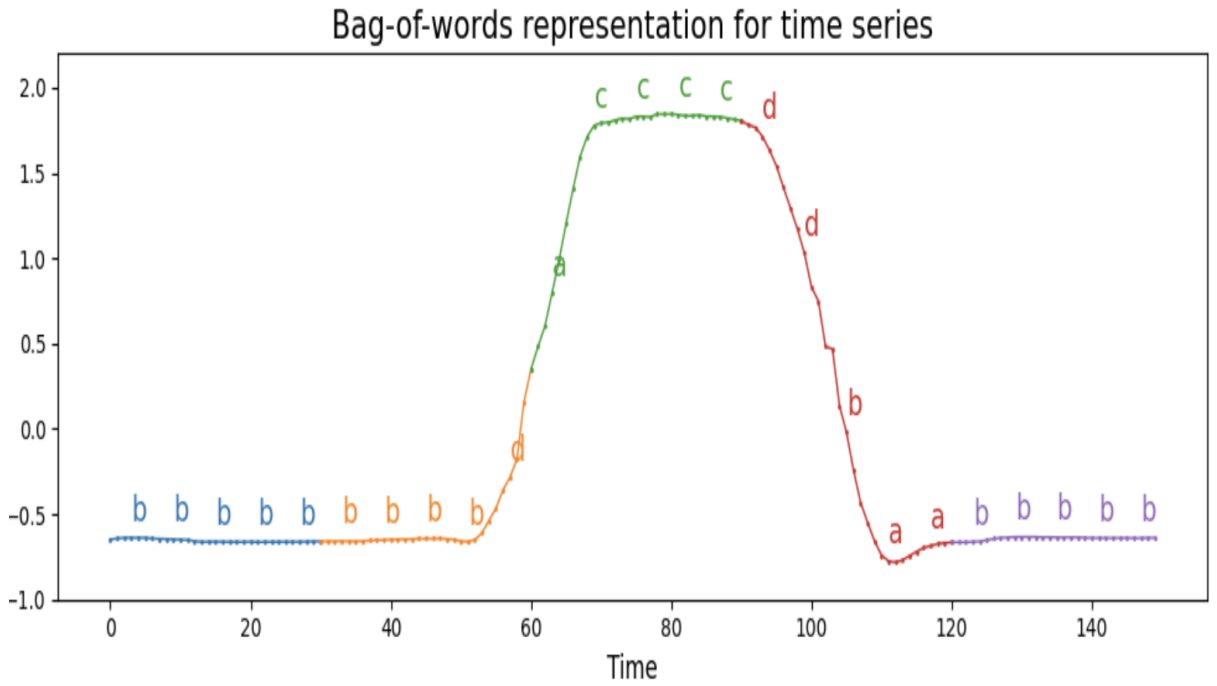
Shapelet Transform

- The Shapelet Transform algorithm extracts shapelets from a data set of time series and returns the distances between the shapelets and the time series.
- A shapelet is defined as a subset of a time series, that is a set of values from consecutive time points.
- The distance between a shapelet and a time series is defined as the minimum of the distances between this shapelet and all the shapelets of same length extracted from this time series.
- The most discriminative shapelets are selected. It is implemented as pyts.transformation.ShapeletTransform



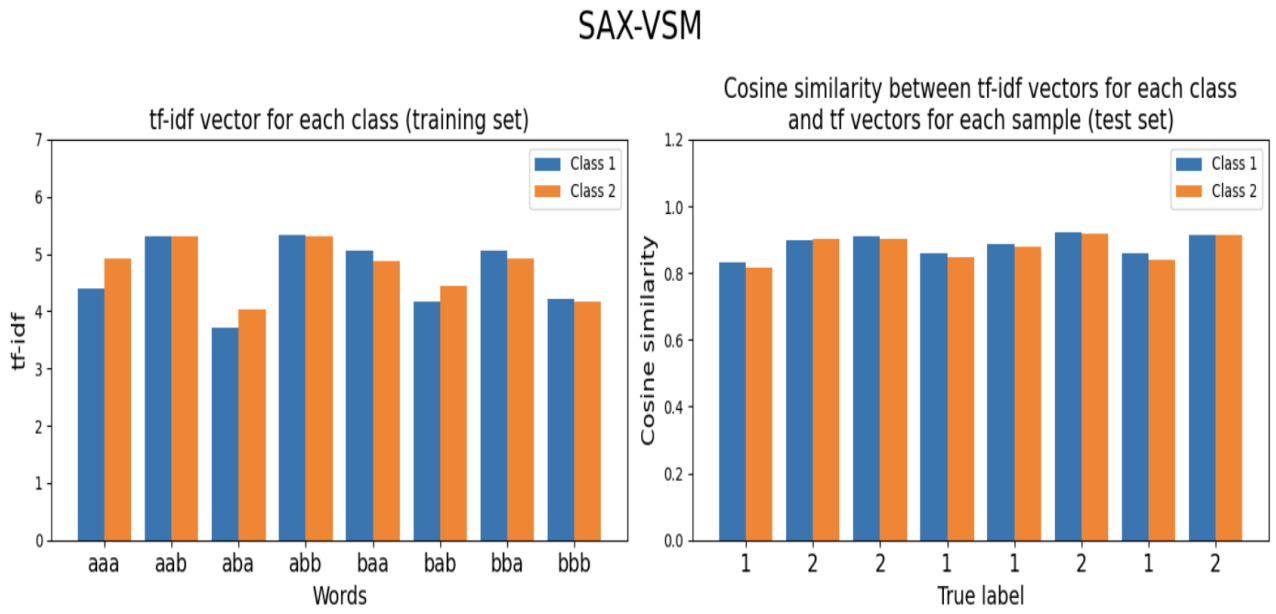
Bag of Words for time series

- Several algorithms for time series classification are based on bag-of-words approaches: a sequence of symbols is transformed into a bag of words.
- BagOfWords extracts subseries using a sliding window, then transforms each subseries into a word using the ***Piecewise Aggregate Approximation*** and ***Symbolic Aggregate approXimation*** algorithms.
- BagOfWords transforms each time series into a bag of words. The sliding window can be controlled with the size and step parameters. The length of each word can be set with size parameter while the ***n_bins*** parameter controls the size of the alphabet to discretize time series.
  The numerosity_reduction parameter controls the removal of all but one occurrence of identical consecutive words.



Symbolic Aggregate approXimation in Vector Space Model (SAX-VSM)

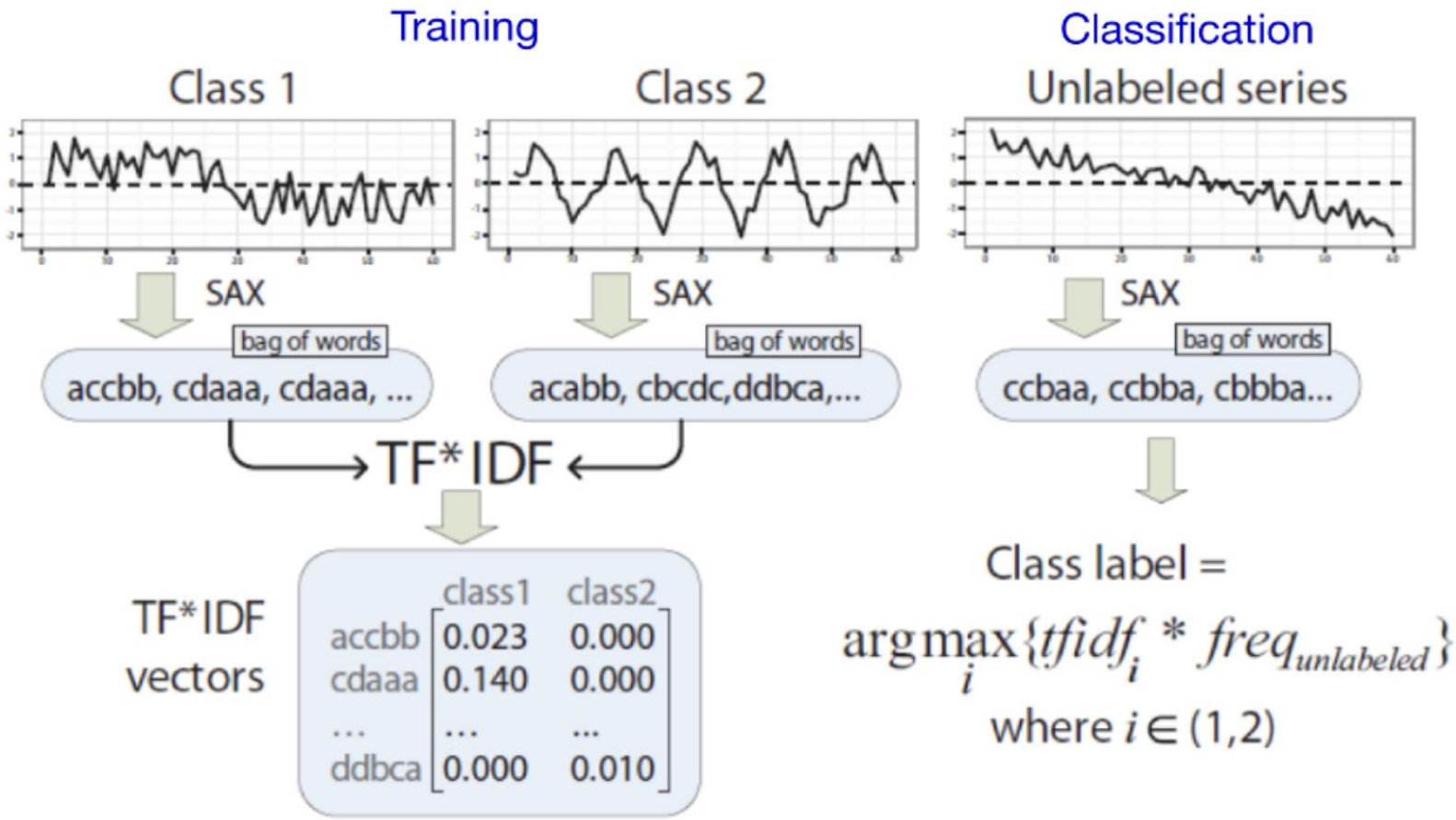
- SAX-VSM is based on two techniques. The first is Symbolic Aggregate approximation, which is a high-level symbolic representation of time series. The second is Vector Space Model based on tf∗idf weighting scheme.
- By using SAX, the algorithm transforms real-valued time series of a single input class into a collection of SAX words (***bag of words***).
- Next, by using tf∗idf weighting, the algorithm transforms these collections (one collection for each of the input classes) into class-characteristic weight vectors, which, in turn, are used in classification built upon Cosine similarity.
- For an unlabeled time series, the predicted label is the label of the tf*idf vector giving the highest cosine similarity with the tf vector of the unlabeled time series



Symbolic Aggregate ApproXimation (SAX)

- SAX-VSM is based on (1)SAX, a high-level symbolic representation of time series (2) Vector Space Model based on tf∗idf weighting scheme.
- SAX discretization performed by: (i) dividing *z*-normalized subsequence into w equal-sized segments, (ii) computing a mean value for each of the segments, and (iii) mapping it to symbols
- Z-normalization, also known as standardization, is a data normalization technique that transforms values to have a mean of 0 and a standard deviation of 1. It's a preprocessing step that's used in machine learning and time series structural pattern mining
- The algorithm consist of two steps: (i) it transforms the original time-series into the PAA representation and (ii) it converts the PAA data into a string
- PAA offers simple and efficient dimensionality reduction – timeseries data has lots of noise
- Z-normalize (Gaussian curve) data before converting to PAA
- Divides the area under *N(0,1)* into *a* equal areas

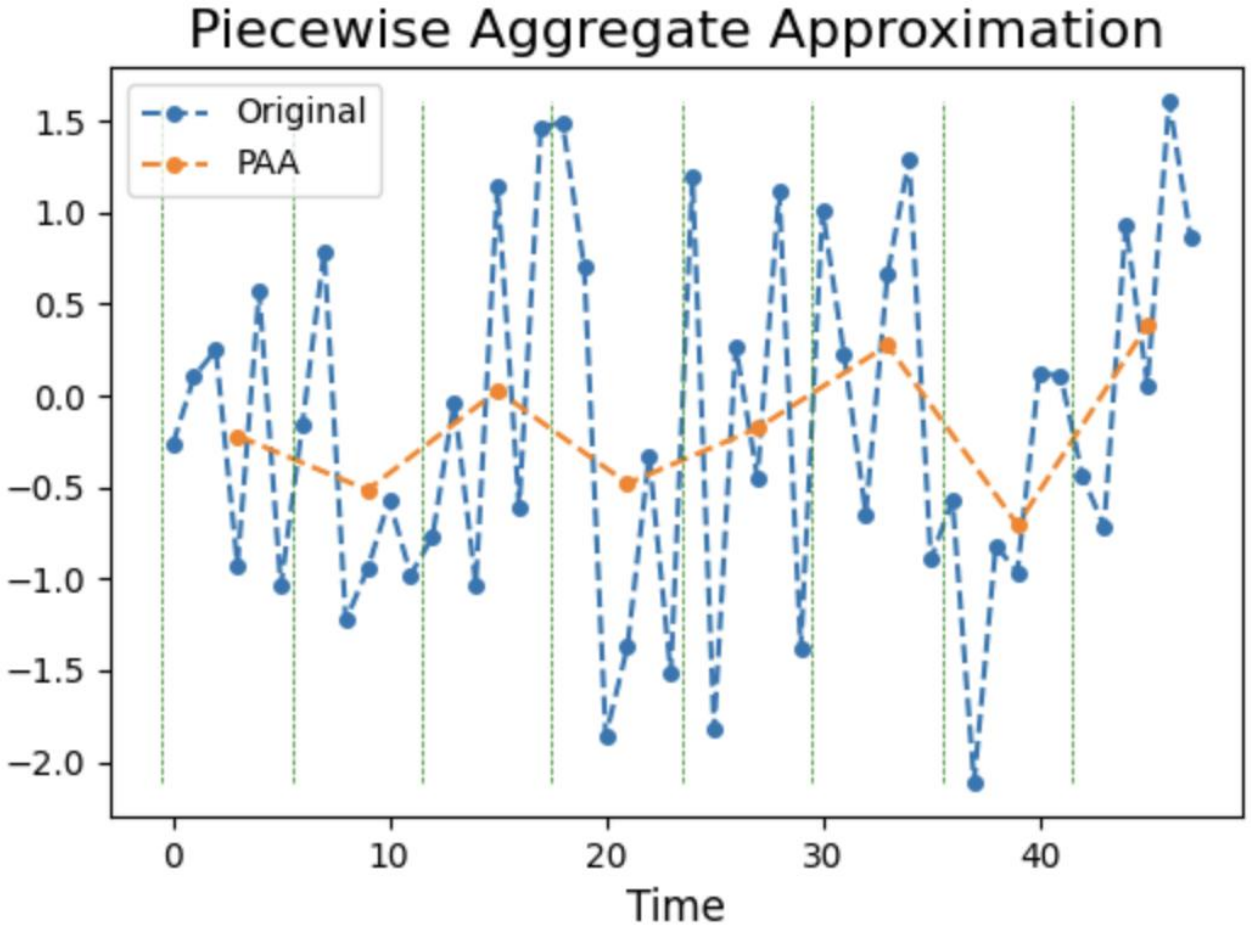
SAX-VSM time series classification algorithm



- Classifier training at the left: all time series of the Class #1 & #2 are converted into a single bag of words– a process which yields two bag of words, one bag per class.
- Next, tf∗idf weighting is applied resulting in the two tf∗idf weight vectors characterizing each of the two classes.
- Classification of an unlabeled time series uses the weight vectors obtained at the training step to compute the similarity score using the Cosine similarity between these vectors and the vector of SAX words frequency obtained by processing the unlabeled input time series with exactly the same SAX discretization parameters used in training. The classification label is assigned by the label of the weight vector which yields the maximal Cosine value.

Piecewise Aggregate Approximation(PAA)

- Time series with a high sampling rate can be very noisy. In order to reduce noise, a technique called ***Piecewise Aggregate Approximation*** is used, consisting in taking the mean over back-to-back points. This decreases the number of points and reduces noise while preserving the trend of the time series.



Piecewise Aggregate Approximation of time series

- PAA approximates a time-series *X* of length *n* into vector $\bar{X}=(\bar{x}1,\ldots,\bar{x}M)$ of any arbitrary length $M \leq n$
- In order to reduce dimensionality from *n* to *M*, we first divide original time-series into *M* equally sized frames
- PAA reduces dimensionality by the mean values of equal sized segments
- PAA downsamples original time series and, in each segment (segments have fixed size), the mean value is retained

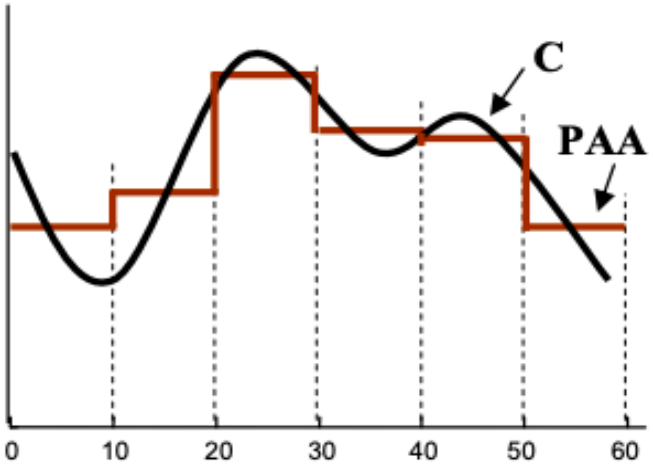


Figure 1: A time series C is represented by PAA (by the mean values of equal segments). In the example above, the dimensionality is reduced from $n = 60$ to $k = 6$.

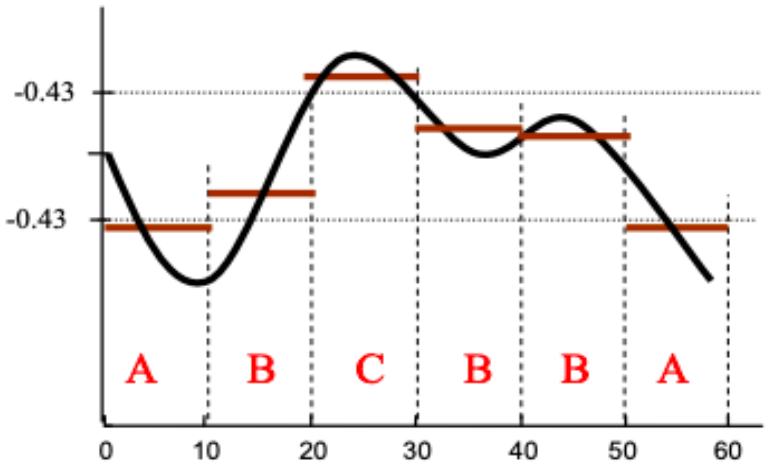
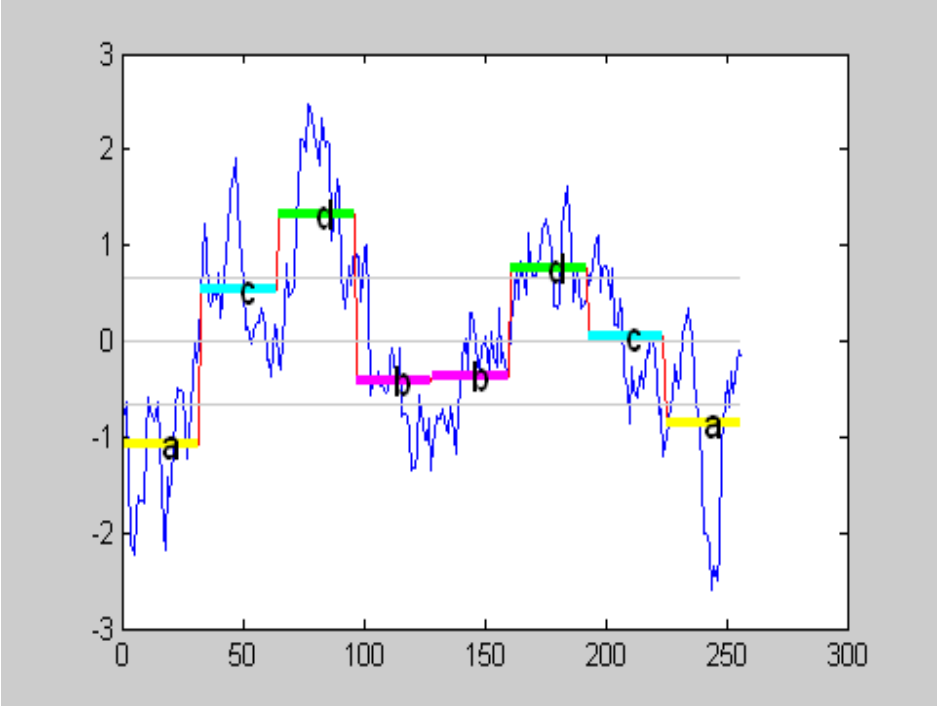
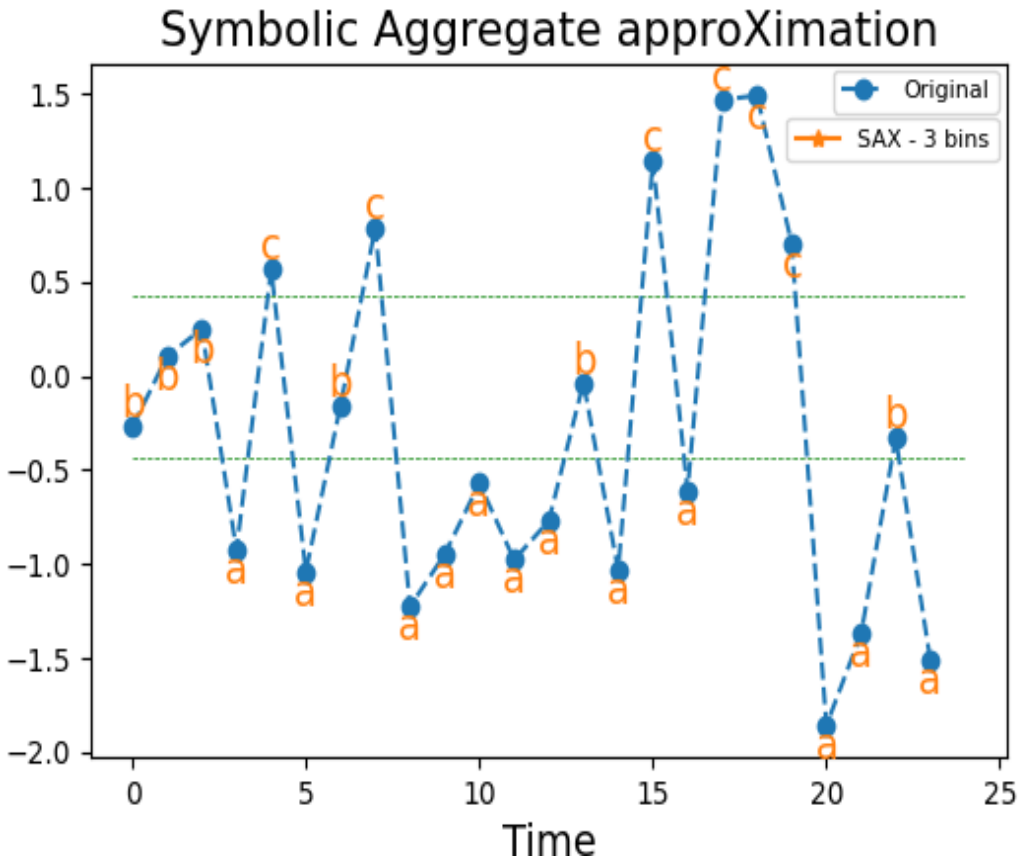


Figure 2: A time series is discretized by SAX. In the example above, with $n = 60$, $k = 6$ and $a = 3$, the time series is mapped to the word *ABCBBA*.

SAX Example

- The following time series is converted to string "*acdbbdca*"

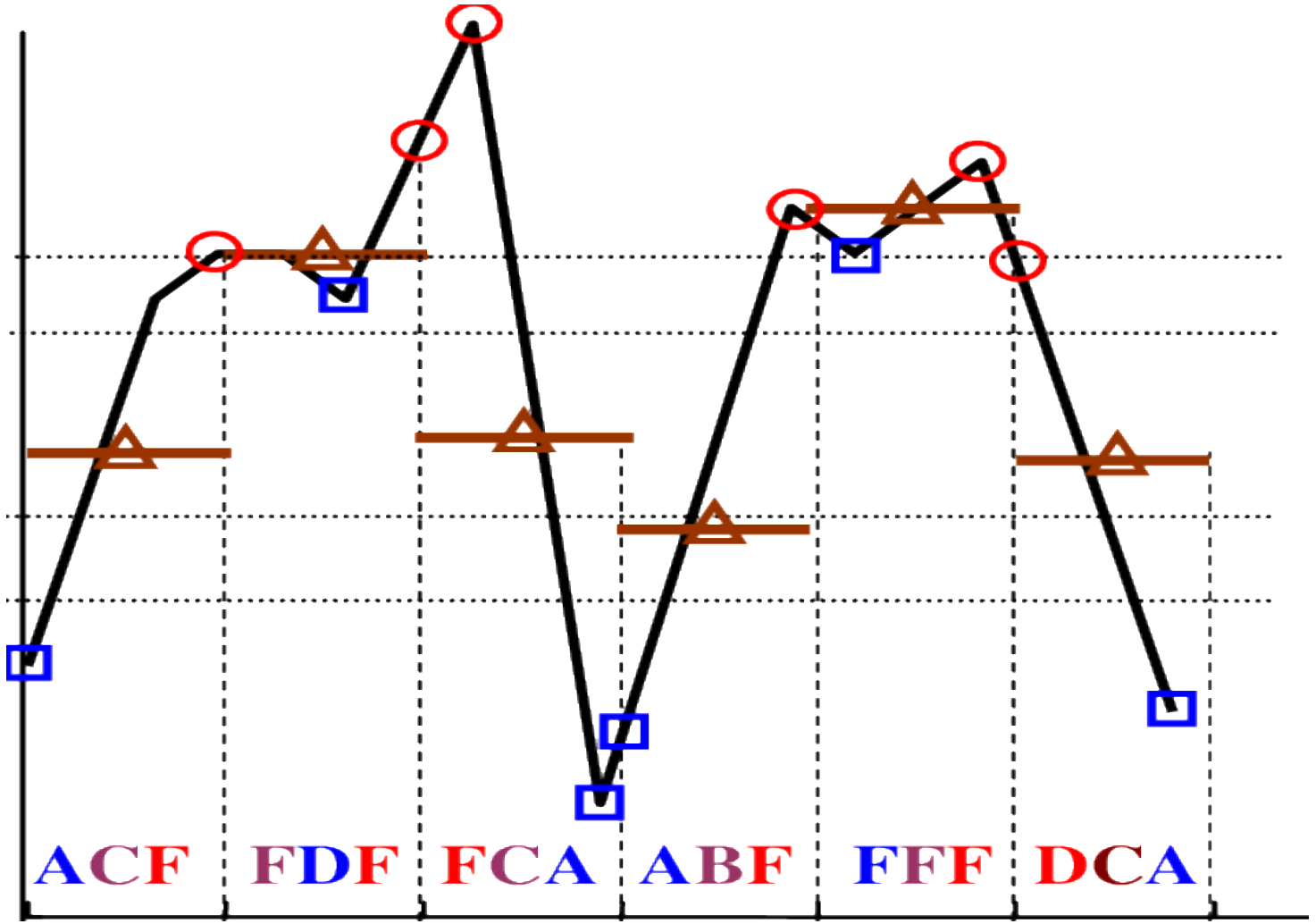




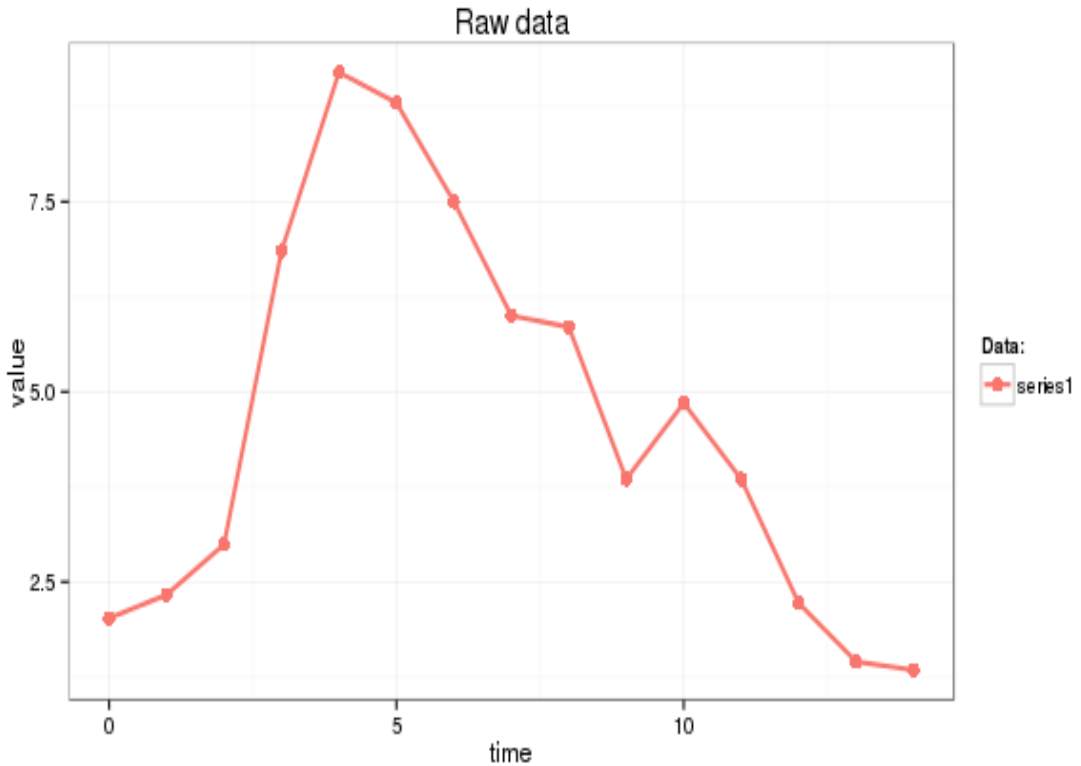
Python SAXVSM Package

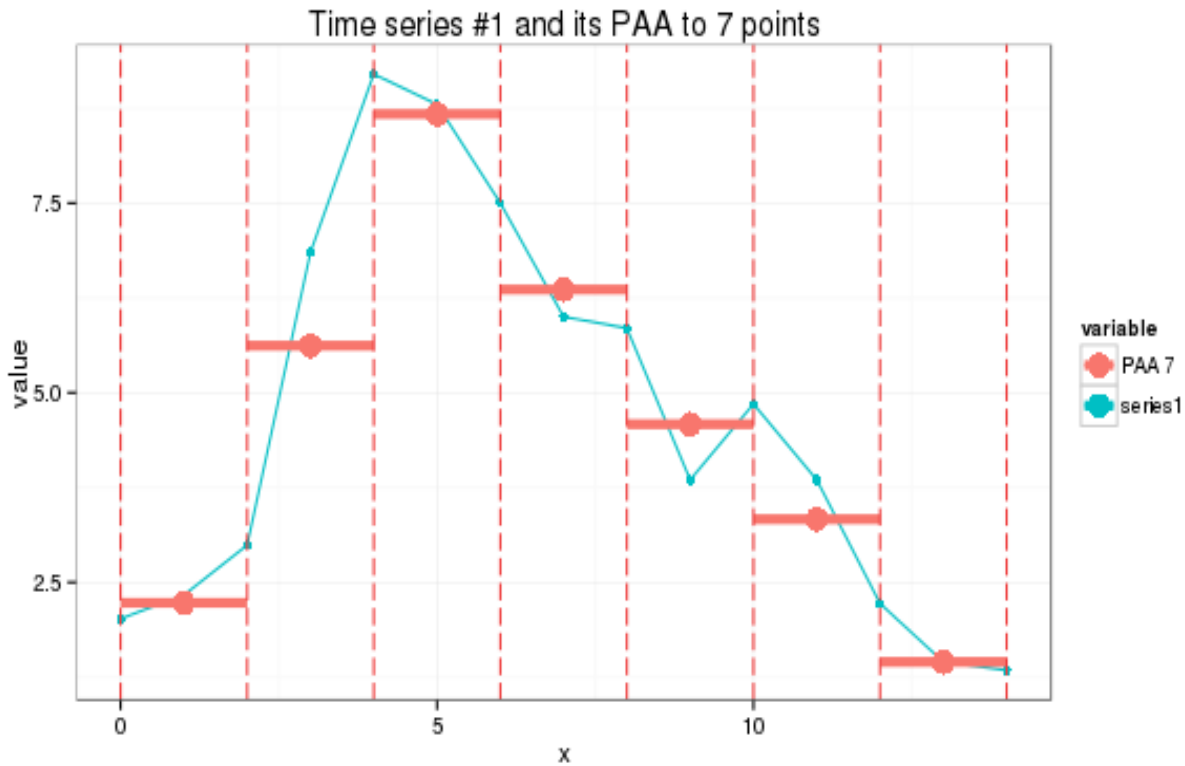
```
>>> from pyts.approximation import   SymbolicAggregateApproximation
>>> X = [[0, 4, 2, 1, 7, 6, 3, 5],
         [2, 5, 4, 5, 3, 4, 2, 3]]
>>> transformer = SymbolicAggregateApproximation()
>>> print(transformer.transform(X))
[['a' 'c' 'b' 'a' 'd' 'd' 'b' 'c’]
['a' 'd' 'c' 'd' 'b' 'c' 'a' 'b']]
```

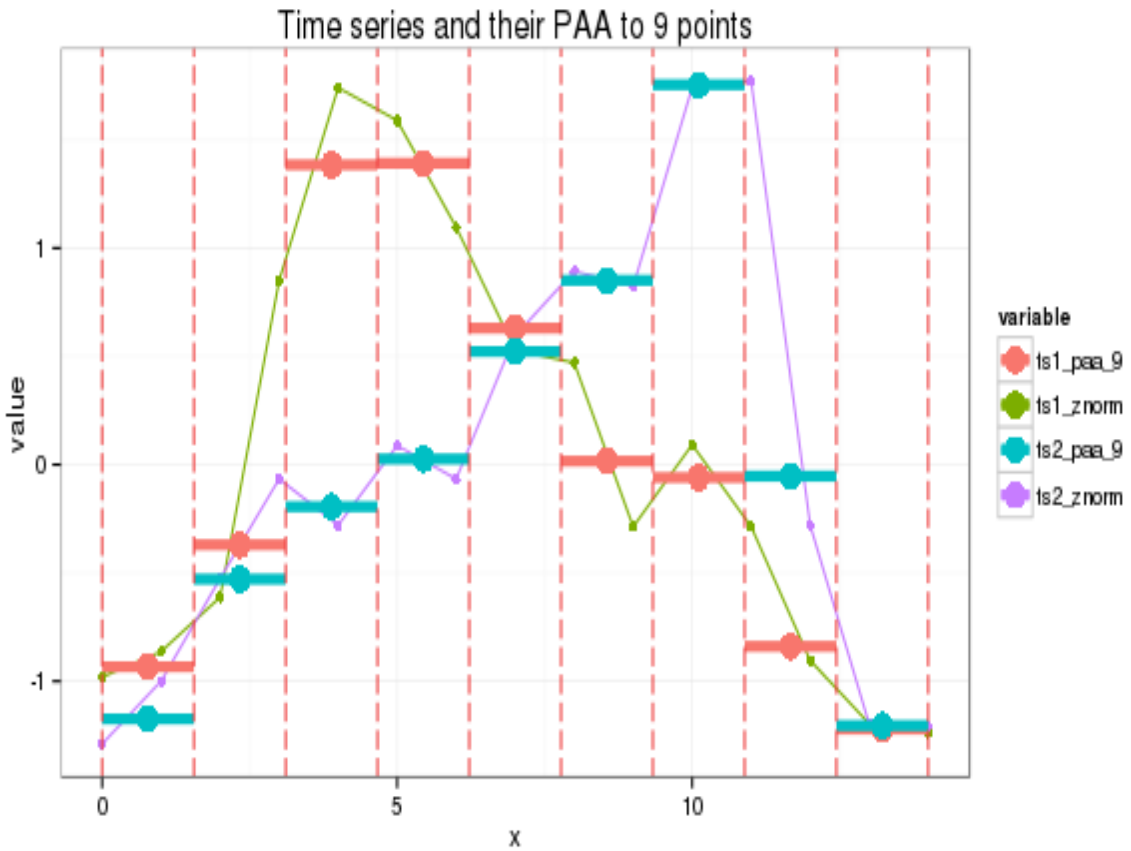
Discretization via Sliding Window

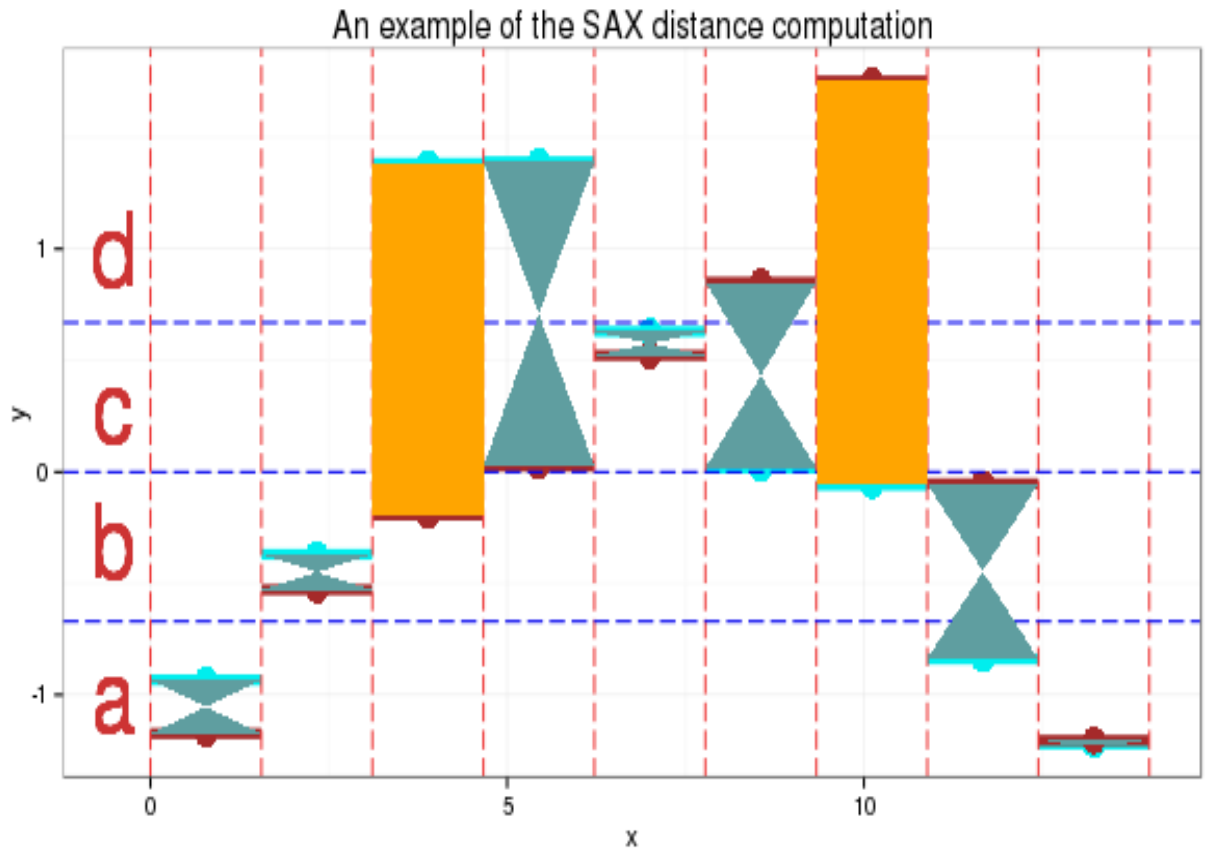


PAA Representation






Time series and their PAA to 9 points
value
x
variable
ts1_paa_9
ts1_znorm
ts2_paa_9
ts2_znorm


An example of the SAX distance computation
y
x
d
c
b
a

SAX Discretization Numerosity Reduction

- Neighboring subsequences extracted via sliding window are often similar to each other
- If same SAX word occurs more than once consecutively, instead of placing every instance into the resulting string, we record only its first occurrence
- ***Numerosity reduction*** is a data reduction technique that reduces the volume of data by representing it in a smaller format. There are two types of numerosity reduction methods: parametric and non-parametric.
- This speeds up the algorithm besides finding variable length motifs and anomalies

$$S_1 = aac_1\ aac_2\ abc_3\ abb_4\ acd_5\ aac_6\ aac_7\ aac_8\ abc_9\ \ldots$$

this process yields:

$$S2 = aac_1\ abc_3\ abb_4\ acd_5\ aac_6\ abc_9$$

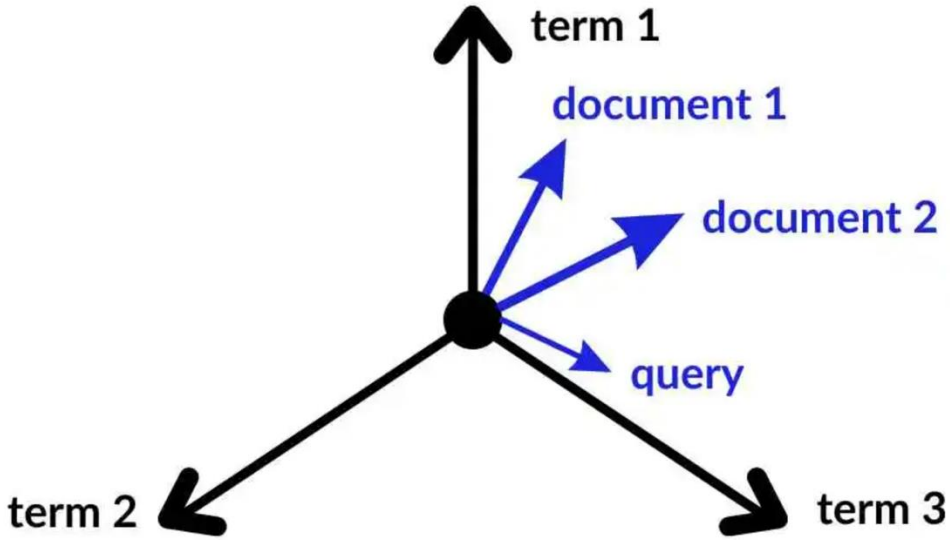
Dimensionality Reduction vs Numerosity Reduction

Dimensionality Reduction	Numerosity Reduction
❖ In dimensionality reduction, data encoding or data transformations are applied to obtain a reduced or compressed form of original data.	❖ In Numerosity reduction, data volume is reduced by choosing suitable alternating forms of data representation.
❖ It can be used to remove irrelevant or redundant attributes.	❖ It is merely a representation technique of original data into smaller form.
❖ In this method, some data can be lost which is irrelevant.	❖ In this method, there is no loss of data.
Methods for dimensionality reduction are: 1. Wavelet transformations. 2. Principal Component Analysis.	**Methods for Numerosity reduction are:** 1. Regression or log-linear model (parametric). 2. Histograms, clustering, sampling (non-parametric).
❖ The components of dimensionality reduction are feature selection and feature extraction.	❖ It has no components but methods that ensure reduction of data volume.
❖ It leads to less misleading data and more model accuracy.	❖ It preserves the integrity of data and the data volume is also reduced.

https://www.geeksforgeeks.org/difference-between-dimensionality-reduction-and-numerosity-reduction/

Vector Space Modelling (tf-idf)

- A vector space contains a collection of objects called vectors which are numerical representations of words, sentence, and even documents. While a simple vector like map coordinates only has two dimensions, those used in natural language processing can have thousands.
- The ***vector space model*** is an algebraic model that represents objects (like text) as vectors. This makes it easy to determine the similarity between words or relevance between a search query and document. Cosine similarity is often used to determine similarity between vectors.
- **tf-idf** stands for *Term frequency-inverse document frequency*
- **tf-idf** is a weighting scheme that assigns each term in a document a weight based on its term frequency (tf) and inverse document frequency (idf)
- The terms with higher weight scores are more important



tf-idf Weighting Scheme

- tf(t, d) = N(t, d), wherein tf(t, d) = term frequency for a term *t* in document *d*
- N(t, d) = number of times a term *t* occurs in document *d*
- *IDF =Log[(# Number of documents) / (Number of documents containing the word)] and*
- *TF = (Number of repetitions of word in a document) / (# of words in a document)*
- IDF, an ***inverse document frequency*** factor diminishes the weight of terms that occur very frequently and increases the weight of terms that occur rarely e.g.(most occurring words like is, as, of etc) in NLP

Cosine Similarity

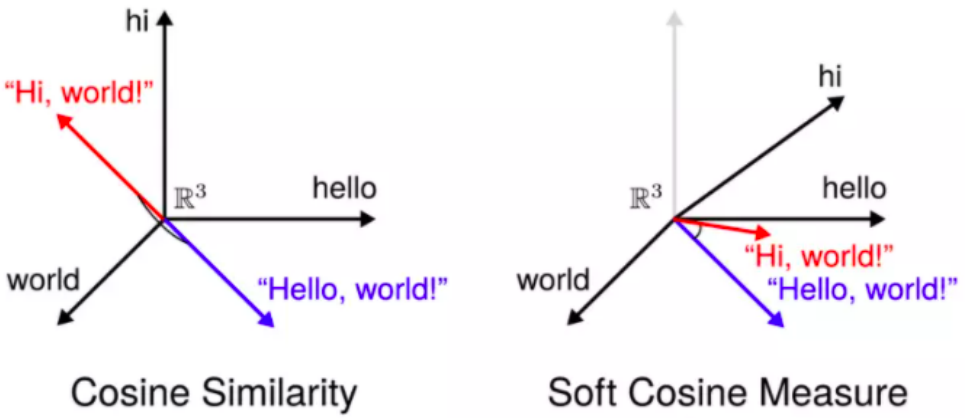
- Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure ***document similarity in text analysis***

- For two vectors ***a*** and ***b*** Cosine similarity is based on their inner product and defined as

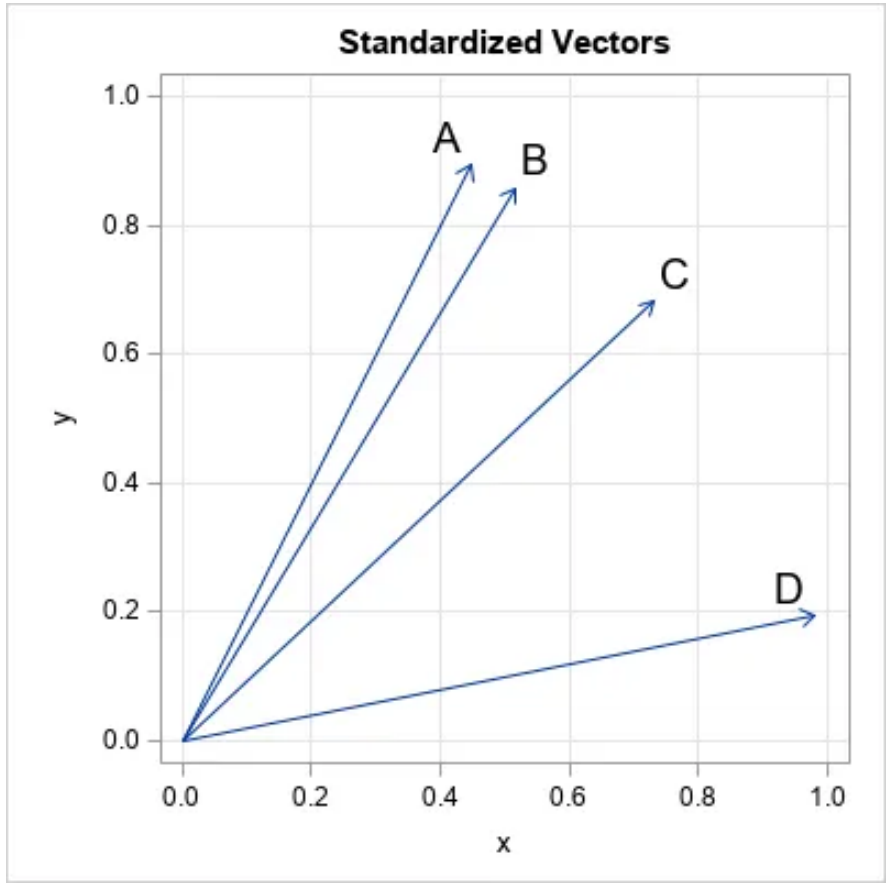
$$\text{similarity}(\boldsymbol{a}, \boldsymbol{b}) = cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{||a|| \cdot ||b||} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}}$$

Cosine Similarity Rules

- Cosine similarity measures the degree to which two vectors point in the same direction
- When vectors point in the same direction, cosine similarity is 1
- when vectors are perpendicular, cosine similarity is 0
- when vectors point in opposite directions, cosine similarity is -1.



Cosine Similarity Soft Cosine Measure



References

- https://www.machinelearningplus.com/nlp/cosine-similarity/
- https://pyts.readthedocs.io/en/stable/auto_examples/classification/plot_saxvsm.html#sphx-glr-auto-examples-classification-plot-saxvsm-py
- https://jmotif.github.io/sax-vsm_site/morea/algorithm/TFIDF.html
- https://medium.com/@peijin/using-sax-vsm-for-financial-time-series-classification-stock-prediction-38b4d10eeb19
- https://jmotif.github.io/sax-vsm_site/morea/algorithm/SAX-VSM.html
- https://blog.minitab.com/en/real-world-quality-improvement/looking-at-past-weather-data-with-minitab-time-series-plots