

Time Series Analysis

Problem Statement

- PH, a tractor and farm equipment manufacturing company, was established a few years after World War II.
- The company has shown a consistent growth in its revenue from tractor sales since its inception. However, over the years the company has struggled to keep it's inventory and production cost down because of variability in sales and tractor demand.

Problem Statement

- The management at PH is under enormous pressure from the shareholders and board to reduce the production cost.
- Additionally, they are also interested in understanding the impact of their marketing and farmer connect efforts towards overall sales.

Problem Statement

- They have hired us as a data science and predictive analytics consultant.
- We will develop an ARIMA model to forecast sale / demand of tractor for next 3 years.
- Additionally, We will also investigate the impact of marketing program on sales by using an exogenous variable ARIMA model.

Problem Statement

- An endogenous variable is one that **is** influenced by other factors in the system. flower growth is affected by sunlight and is therefore endogenous.

Exogenous variables...

- are fixed when they enter the model.
- are taken as a “given” in the model.
- influence endogenous variables in the model.
- are not determined by the model.
- are not explained by the model.

Problem Statement

- As a part of the project, one of the production units we are analyzing is based in South East Asia.
- This unit is completely independent and caters to neighboring geographies. This unit is just a decade and a half old. In 2014 , they captured 11% of the market share, a 14% increase from the previous year.

Problem Statement

- However, being a new unit they have very little bargaining power with their suppliers to implement Just-in-Time (JiT) manufacturing principles that have worked really well in PH's base location.
- Hence, they want to be on top of their production planning to maintain healthy business margins.

Problem Statement

- Monthly sales forecast is the first step we have suggested to this unit towards effective inventory management.
- The MIS team shared the month on month (MoM) sales figures (number of tractors sold) for the last 12 years

Time Series Analysis

- Time series analysis is extensively used to forecast company sales, product demand, stock market trends, agricultural production etc.
- The fundamental idea for time series analysis is to decompose the original time series (sales, stock market trends, etc.) into several independent components.

Time Series Analysis

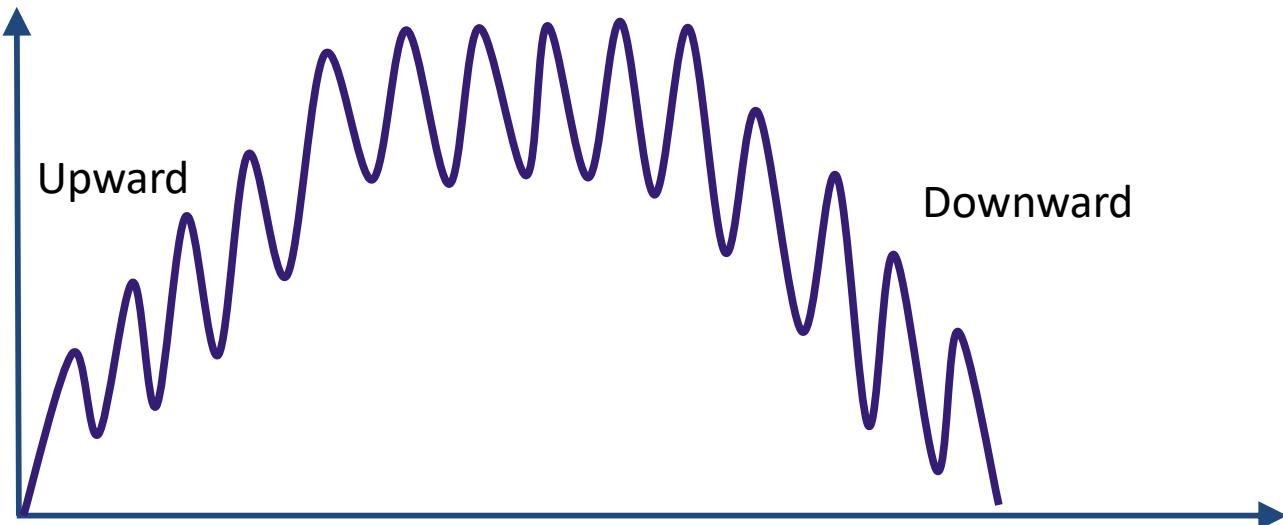
- Typically, business time series are divided into the following four components:
- **Trend** – overall direction of the series i.e. upwards, downwards etc.
- **Seasonality** – monthly or quarterly patterns
- **Cycle** – long-term business cycles, they usually come after 5 or 7 years
- **Irregular remainder** – random noise left after extraction of all the components

Time Series Analysis

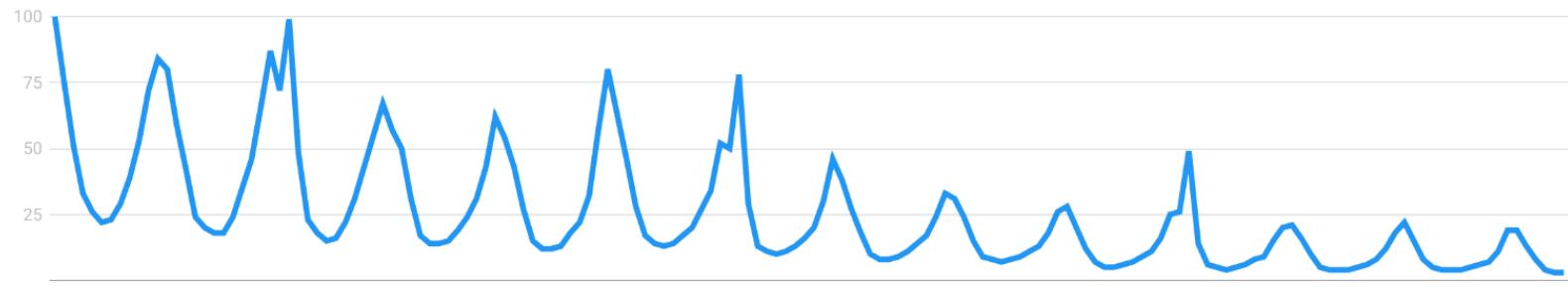
- Interference of these components produces the final series.
- Why decomposing the original / actual time series into components?
- It is much easier to forecast the individual regular patterns produced through decomposition of time series than the actual series.

- Trends

Horizontal/Stationary



- Seasonality - Repeating trends



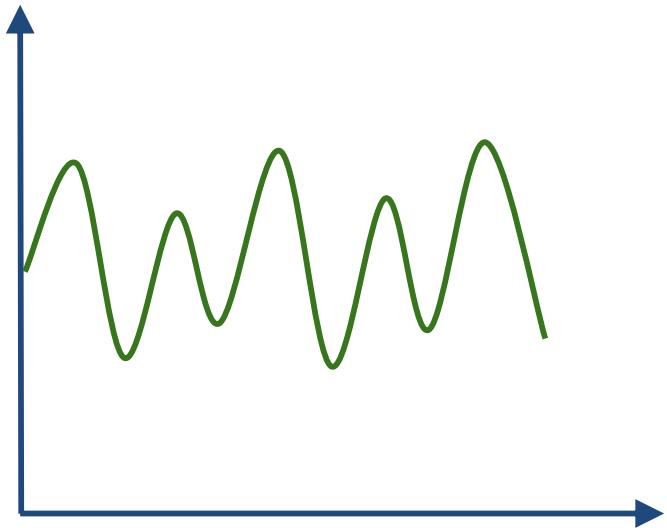
Google Trends - “Snowboarding”

- Cyclical - Trends with no set repetition.

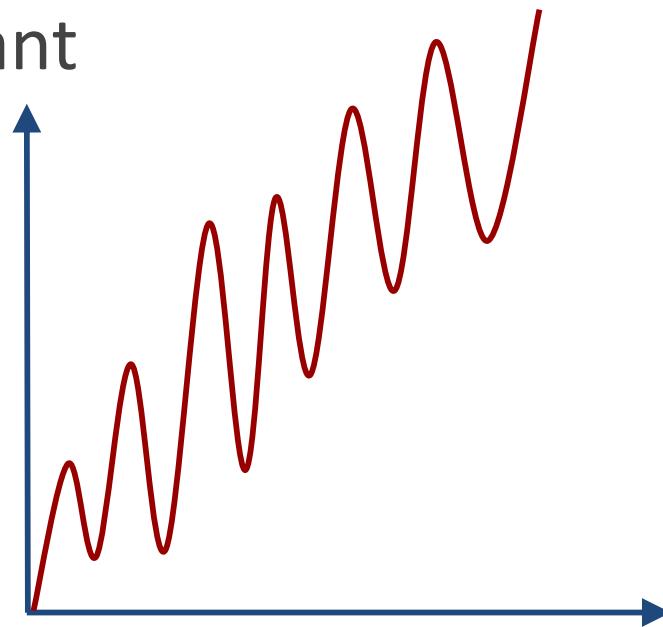


- Stationary vs Non-Stationary Data
 - To effectively use ARIMA, we need to understand Stationarity in our data.
 - So what makes a data set Stationary?
 - A Stationary series has constant mean and variance over time.

- Mean needs to be constant

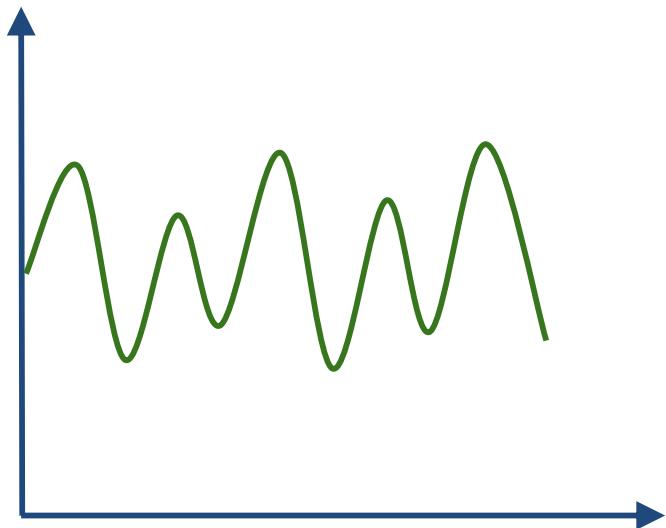


Stationary

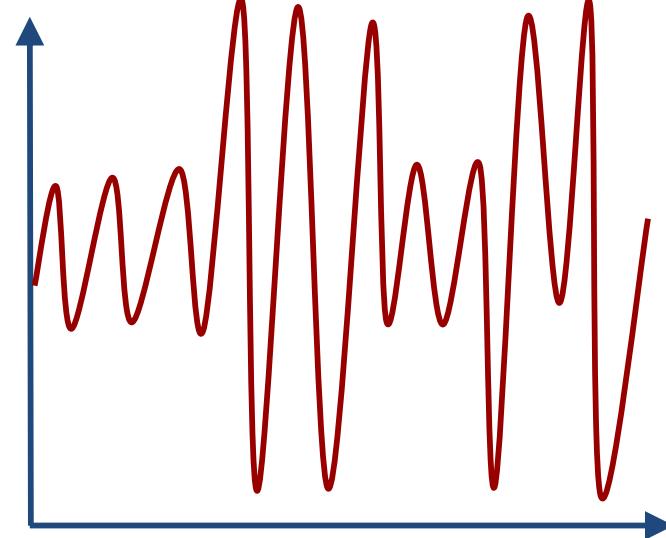


Non-Stationary

- Variance should not be a function of time



Stationary



Non-Stationary

- A Stationary data set will allow our model to predict that the mean and variance will be the same in future periods.

- There are also mathematical tests you can use to test for stationarity in your data.
- A common one is the Augmented Dickey–Fuller test

- If we've determined your data is not stationary (either visually or mathematically), we will then need to transform it to be stationary in order to evaluate it and what type of ARIMA terms you will use.

- One simple way to do this is through “differencing”.

Original Data

Time 1	10
Time 2	12
Time 3	8
Time 4	14
5	

First Difference

Time 1	NA
Time 2	2
Time 3	-4
Time 4	6
Time 5	-7

Second Difference

Time 1	NA
Time 2	NA
Time 3	-6
Time 4	10
Time 5	-13

- You can continue differencing until you reach stationarity (which you can check visually and mathematically)
- Each differencing step comes at the cost of losing a row of data.

- For seasonal data, we can also difference by a season.
- For example, if we had monthly data with yearly seasonality, we could difference by a time unit of 12, instead of just 1.

- With our data now stationary it is time the p,d,q terms and how we choose them.
- A big part of this are AutoCorrelation Plots and Partial AutoCorrelation Plots.

Trend

- From the plots it is obvious that there is some kind of increasing trend in the series along with seasonal variation.
- Stationarity is a vital assumption we need to verify if our time series follows a stationary process or not.

Trend

- We can do by
 - Plots: review the time series plot of our data and visually check if there are any obvious trends or seasonality
 - Statistical tests: use statistical tests to check if the expectations of stationarity are met or have been violated.

Trend using MAs

- Moving averages over time
 - One way to identify a trend pattern is to use moving averages over a specific window of past observations.
 - This smoothens the curve by averaging adjacent values over the specified time horizon (window).

Seasonality

- People tend to go on vacation mainly during summer holidays.
- At some time periods during the year people tend to use aircrafts more frequently. We can check the hypothesis of a seasonal effect

Noise

- To understand the underlying pattern in the number of international airline passengers, we assume a multiplicative time series decomposition model
- Purpose is to understand underlying patterns in temporal data to use in more sophisticated analysis like Holt-Winters seasonal method or ARIMA.

Noise

- Noise - is the residual series left after removing the trend and seasonality components

Stationarize a Time series

- Before models forecasting can be applied, the series must be transformed into a stationary time series.
- The Augmented-Dickey Fuller Test can be used to test whether or not a given time series is stationary.

Stationarize a Time series

- If the test statistic is smaller than the critical value, the hypothesis is rejected, the series would be stationary, and no further transformations of the data would be required.

Residuals Serial Correlation

- When the residuals (errors) in a time series are correlated with each other it is said to exhibit serial correlation.
- Autocorrelation is a better measurement for the dependency structure, because the autocovariacne will be affected by the underlying units of measurement for the observation.

White Noise & ACF & PACF

- Random process is white noise process
- Errors are serially uncorrelated if they are independent and identically distributed (iid).
- It is important because if a time series model is successful at capturing the underlying process, residuals of the model will be iid and resemble a white noise process.

White Noise

- Part of time series analysis is simply trying to fit a model to a time series such that the residual series is indistinguishable white noise.

ACF & PACF

- The plots of the Autocorrelation function (ACF) and the Partial Autorrelation Function (PACF) are the two main tools to examine the time series dependency structure.
- The ACF is a function of the time displacement of the time series itself.
- It is the similarity between observations as a function of the time lag between them.

PACF

- The PACF is the conditional correlation between two variables under the assumptions that the effects of all previous lags on the time series are known.

Random Walk

- What is special about the random walk is, that it is non-stationary, that is, if a given time series is governed by a random walk process it is unpredictable.
- It has high ACF for any lag length
- The normal QQ plot and the histogram indicate that the series is not normally distributed

Random Walk

- The random walk is a first order autoregressive process that is, this causes the process to be non-stationary.
- The process can be made stationary

Auto Regressive Model – AR(p)

- The random walk process belongs to a more general group of processes, called autoregressive process
- The current observation is a linear combination of past observations.
- An AR(1) time series is one period lagged weighted version of itself.

The Moving Average Model - MA(q)

- The moving average model MA(q) assumes that the observed time series can be represented by a linear combination of white noise error terms.
- The time series will always be stationary.

ARIMA Forecasting

- An autoregressive integrated moving average (ARIMA) model is an generalization of an autoregressive moving average (ARMA) model.
- Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting).

ARIMA Forecasting

- ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.
- There are three parameters (p, d, q) that are used to parametrize ARIMA models. Hence, an ARIMA model is denoted as $\text{ARIMA}(p, d, q)$
- Each of these three parts is an effort to make the time series stationary, i. e. make the final residual a white noise pattern.

Box-Jenkins Approach to non-Seasonal ARIMA Modeling

- In time series analysis, the Box–Jenkins method,[1] named after the statisticians George Box and Gwilym Jenkins, applies autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) models to find the best fit of a time-series model to past values of a time series.

Box-Jenkins Approach to non-Seasonal ARIMA Modeling

- The original model uses an iterative three-stage modeling approach:
- Model identification and model selection:

Box-Jenkins Approach to non-Seasonal ARIMA Modeling

- Making sure that the variables are stationary, identifying seasonality in the dependent series (seasonally differencing it if necessary), and using plots of the autocorrelation and partial autocorrelation functions of the dependent time series to decide which (if any) autoregressive or moving average component should be used in the model.

Box-Jenkins Approach to non-Seasonal ARIMA Modeling

- Parameter estimation using computation algorithms to arrive at coefficients that best fit the selected ARIMA model.
- Model checking by testing whether the estimated model conforms to the specifications of a stationary univariate process.

Box-Jenkins Approach to non-Seasonal ARIMA Modeling

- The residuals should be independent of each other and constant in mean and variance over time.
- If the estimation is inadequate, we have to return to step one and attempt to build a better model.

Optimal Parameter Selection

- To fit the time series data to a seasonal ARIMA model with parameters $\text{ARIMA}(p, d, q)(P, D, Q)s$ the optimal parameters need to be found first.
- This is done via grid search, the iterative exploration of all possible parameters constellations.

Optimal Parameter Selection

- Depending on the size of the model parameters $\$(p, d, q)(P, D, Q)s\$$ this can become an extremely costly task with regard to computation. We start of by generating all possible parameter constellation we'd like to evaluate.

Akaike Information Criterion (AIC).

- For all possible parameter constellations from both lists pdq and seasonal_pdq the algorithm will create a model and eventually pick the best one to proceed.
- The best model is chosen based on the Akaike Information Criterion (AIC).

Akaike Information Criterion (AIC).

- The Akaike information criterion (AIC) is a measure of the relative quality of statistical models for a given set of data.
- Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Hence, AIC provides a means for model selection.
- It measures the trade-off between the goodness of fit of the model and the complexity of the model (number of included and estimated parameters).

One step ahead prediction

- The `get_prediction` and `conf_int` methods calculate predictions for future points in time for the previously fitted model and the confidence intervals associated with a prediction, respectively.
- The `dynamic=False` argument causes the method to produce a one-step ahead prediction of the time series

MSE

- To quantify the accuracy between model fit and true observations we use the mean squared error (MSE).
- The MSE computes the squared difference between the true and predicted value.

Out of sample Prediction

- To put the model to the real test with a 24-month-head prediction.
- This requires to pass the argument `dynamic=False` when using the `get_prediction` method."

Long term forecasting

- Finally, a 10 year ahead forecast, leveraging a seasonal ARIMA model trained on the complete time series y .
- Grid search found the best model to be of form SARIMAX(2, 1, 3)(1, 2, 1)12 for the data vector y .

Tractor Sales & Marketing Analysis

PH Trend - Time Series Decomposition

- Remove wrinkles from our time series using moving average.
- Moving average of different time periods i.e. 4,6,8, and 12 months

PH Tractor - Dicky Fuller Test on the timeseries

- Run the Dicky Fuller Test on the timeseries and
- Verify the null hypothesis that the TS is non-stationary.

PH Tractor Seasonality Time Series Decomposition 1/2

- The first thing to do is to see how number of tractors sold vary on a month on month basis. .
- A stacked annual plot to observe seasonality
- A box plot

PH Tractor Seasonality Time Series Decomposition 2/2

- The tractor sales have been increasing without fail every year.
- July and August are the peak months for tractor sales and the variance and the mean value in July and August are also much higher than any of the other months.
- We can see a seasonal cycle of 12 months where the mean value of each month starts with a increasing trend in the beginning of the year and drops down towards the end of the year.
- We can see a seasonal effect with a cycle of 12 months.

PH Tractor Irregular Remainder 1/4

- To decipher underlying patterns in tractor sales, we build a multiplicative time series decomposition model
- The primary purpose is to understand underlying patterns in temporal data to use in more sophisticated analysis like Holt-Winters seasonal method or ARIMA.

PH Tractor Irregular Remainder 2/4

- Key observations:
- 1) Trend: 12-months moving average looks similar to a straight line hence we could use linear regression to estimate the trend in this data.
- 2) Seasonality: seasonal plot displays a fairly consistent month-on-month pattern. The monthly seasonal components are average values for a month after removal of trend. Trend is removed from the time series

PH Tractor Irregular Remainder 3/4

- Key observations:
- 3) Irregular Remainder (random): is the residual left in the series after removal of trend and seasonal components.
- The expectations from remainder component is that it should look like a white noise i.e. displays no pattern at all.

PH Tractor Irregular Remainder 4/4

- Key observations:
- However, for our series residual displays some pattern with high variation on the edges of data i.e. near the beginning (2004-07) and the end (2013-14) of the series.

PH ARIMA Modeling 1/4

- ARIMA is a combination of 3 parts i.e. AR (AutoRegressive), I (Integrated), and MA (Moving Average). A convenient notation for ARIMA model is ARIMA(p,d,q).
- Here p,d, and q are the levels for each of the AR, I, and MA parts.
- Each of these three parts is an effort to make the final residuals display a white noise pattern (or no pattern at all).

PH ARIMA Modeling 2/4

- In each step of ARIMA modeling, time series data is passed through these 3 parts like a sugar cane through a sugar cane juicer to produce juice-less residual. The sequence of three passes for ARIMA analysis is as following:
- 1st Pass of ARIMA to Extract Juice / Information
- Integrated (I) – subtract time series with its lagged series to extract trends from the data

PH ARIMA Modeling 3/4

- In this pass of ARIMA juicer, we extract trend(s) from the original time series data.
- Differencing is one of the most commonly used mechanisms for extraction of trends. Here, the original series is subtracted with it's lagged series e.g. November's sales values are subtracted with October's values to produce trend-less residual series.

PH ARIMA Modeling 4/4

- The formulae for different orders of differencing in the plot a time series data with a linearly upward trend is displayed.
- Adjacent to that plot is the 1st order differenced plot for the same data.
- We can notice after 1st order differencing, trend part of the series is extracted and the difference data (residual) does not display any trend.

2nd Pass of ARIMA 1/2

- AutoRegressive (AR) – extract the influence of the previous periods' values on the current period
- After the time series data is made stationary through the integrated (I) pass, the AR part of the ARIMA juicer gets activated.
- As the name auto-regression suggests, here we try to extract the influence of the values of previous periods on the current period e.g. the influence of the September and October's sales value on the November's sales.

2nd Pass of ARIMA 2/2

- This is done through developing a regression model with the time lagged period values as independent or predictor variables.
- AR model of order 1 i.e. $p=1$ or $\text{ARIMA}(1,0,0)$ is represented

3rd Pass of ARIMA 1/2

- Moving Average (MA) – extract the influence of the previous period's error terms on the current period's error
- Finally, the last component of ARIMA i.e. MA involves finding relationships between the previous periods' error terms on the current period's error term.

3rd Pass of ARIMA 2/2

- Moving Average (MA) part of ARIMA is developed with the simple multiple linear regression values with the lagged error values as independent or predictor variables.
- MA model of order 1 i.e. $q=1$ or ARIMA(0,0,1) is represented

White Noise & ARIMA 1/3

- White noise is a funny thing, if we look at it for long we will start seeing some false patterns. This is because the human brain is wired to find patterns, and at times confuses noises with signals.
- The biggest proof of this is how people lose money every day on the stock market.

White Noise & ARIMA 2/3

- This is the reason why we need a mathematical or logical process to distinguish between a white noise and a signal (juice / information).
- Consider the simulated white noise

White Noise & ARIMA 3/3

- Key observations:
 - If we stare at the graph for a reasonably long time we may start seeing some false patterns
 - A good way to distinguish between signal and noise is ACF (AutoCorrelation Function). This is developed by finding the correlation between a series of its lagged values..

ACF Plot 1/2

- Key observations:
 - We could see that for lag = 0 the ACF plot has the perfect correlation i.e. $\rho = 1$. because any data with itself will always have the perfect correlation
 - However as expected, our white noise doesn't have a significant correlation with its historic values ($lag \geq 1$).
 -

ACF Plot 2/2

- Key observations:
 - The dotted horizontal lines in the plot show the threshold for the insignificant region i.e. for a significant correlation the vertical bars should fall outside the horizontal dotted lines.

Step 2: Difference data to make data stationary on mean (remove trend) 1/2

- Key Observations:
 - The above series is not stationary on variance i.e. variation in the plot is increasing as we move towards the right of the chart.
 - We need to make the series stationary on variance to produce reliable forecasts through ARIMA models.

Step 2: Difference data to make data stationary on mean (remove trend) 2/2

- Key Observations:
 - The tractor sales has an upward trend for tractors sales and there is also a seasonal component.
 - Make the series stationary by removing the upward trend through 1st order differencing of the series

Step 3: log transform data to make data stationary on variance

- One of the best ways to make a series stationary on variance is through transforming the original series through log transform.
- Log transform original tractor sales series it to make it stationary on variance.
- This series is not stationary on mean since we are using the original data without differencing. But now the series looks stationary on variance.

Step 4: Difference log transform data to make data stationary on both mean and variance 1/2

- Look at the differenced plot for log transformed series to reconfirm if the series is actually stationary on both mean and variance.

Step 4: Difference log transform data to make data stationary on both mean and variance 2/2

- Key Observations:
 - This series looks stationary on both mean and variance.
 - This also gives us the clue that I or integrated part of our ARIMA model will be equal to 1 as 1st difference is making the series stationary.

Step 5: Plot ACF and PACF to identify potential AR and MA model 1/2

- Key Observations:
- Since, there are enough spikes in the plots outside the insignificant zone (dotted horizontal lines) we can conclude that the residuals are not random.
- This implies that there is patterns or information available in residuals to be extracted by AR and MA models.

Step 5: Plot ACF and PACF to identify potential AR and MA model 2/2

- Key Observations:
 - Also, there is a seasonal component available in the residuals at the lag 12 (represented by spikes at lag 12).
 - Since we are analyzing monthly data that tends to have seasonality of 12 months because of patterns in tractor sales.

Step 6: Identification of best fit ARIMA model 1/2

- In order to fit the time series data with a seasonal ARIMA model, we need to first find the values of ARIMA(p,d,q)(P,D,Q)s that optimize a metric of interest such as AIC or BIC.
- We generate combination of p,d and q to select the optimal parameter values for our ARIMA(p,d,q)(P,D,Q)s time series model.

Step 6: Identification of best fit ARIMA model 2/2

- This technique is known as "grid search" where we iteratively explore different combinations of parameters.
- For each such combination of parameters, we try to fit a new seasonal ARIMA model with the SARIMAX() function from the statsmodels module and assess AIC or BIC score.
- The model with the best score wins and the parameters for that model are the optimal parameters.

AIC & BIC 1/4

- The best fit model is selected based on Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC) values.
- The idea is to choose a model with minimum AIC and BIC values.
- Akaike Information Criterion (AIC) - AIC is an effort to balance the model between goodness-of-fit and number of parameters used in the model.

AIC & BIC 2/4

- Bayesian Information Criterion (BIC) is another variant of AIC and is used for the same purpose of best fit model selection.
- For the best possible model selection, we look at AIC, BIC, and AICc (AIC with sample correction) if all these values are minimum for a given model.

AIC & BIC 3/4

- As expected, our model has I (or integrated) component equal to 1.
- This represents differencing of order 1. There is additional differencing of lag 12 in the above best fit model.

AIC & BIC 4/4

- Moreover, the best fit model has MA value of order 1.
Also, there is seasonal MA with lag 12 of order 1.

Step 7: Predict sales on in-sample date using the best fit ARIMA model

- The next step is to predict tractor sales for in-sample data and find out how close is the model prediction on the in-sample data to the actual truth.

Step 8: Forecast sales using the best fit ARIMA model 1/2

- The next step is to predict tractor sales for next 3 years i.e. for 2015, 2016, and 2017 through the above model.
- Get forecast 36 steps (3 years) ahead in future

Step 8: Forecast sales using the best fit ARIMA model 2/2

- Key Observations:
 - A short-term forecasting model, say a couple of business quarters or a year, is usually a good idea to forecast with reasonable accuracy.
 - A long-term model like the one above needs to evaluated on a regular interval of time (say 6 months).
 - The idea is to incorporate the new information available with the passage of time in the model.

Step 9: Plot ACF and PACF for residuals of ARIMA 1/6

- Plot ACF and PACF for residuals of ARIMA model to ensure no more information is left for extraction
- Finally, create an ACF and PACF plot of the residuals of our best fit
- ARIMA model i.e. ARIMA(0,1,1)(1,0,1)[12].

Step 9: Plot ACF and PACF for residuals of ARIMA 2/6

- Key Observations:
 - We need to ensure that the residuals of our model are uncorrelated and normally distributed with zero-mean.
 - If it is not that it signifies that the model can be further improved and we repeat the process with the residuals.

Step 9: Plot ACF and PACF for residuals of ARIMA 3/6

- Key Observations:
 - In this case, our model diagnostics suggests that the model residuals are normally distributed based on the following:
 - The KDE plot of the residuals on the top right is almost similar with the normal distribution.

Step 9: Plot ACF and PACF for residuals of ARIMA 4/6

- Key Observations:
 - The qq-plot on the bottom left shows that the ordered distribution of residuals (blue dots) follows the linear trend of the samples taken from a standard normal distribution with $N(0, 1)$.
 - This is a strong indication that the residuals are normally distributed.

Step 9: Plot ACF and PACF for residuals of ARIMA 5/6

- Key Observations:
- The residuals over time (top left plot) don't display any obvious seasonality and appear to be white noise.
 - This is confirmed by the autocorrelation (i.e. correlogram) plot on the bottom right, which shows that the time series residuals have low correlation with lagged versions of itself.

Step 9: Plot ACF and PACF for residuals of ARIMA 6/6

- Key Observations:
- Those observations coupled with the fact that there are no spikes outside the insignificant zone for both ACF and PACF plots lead us to conclude that residuals are random with no information or patterns in them and our model produces a satisfactory fit that could help us understand our time series data and forecast future values.
- It means that our ARIMA model is working fine.

Sales & Marketing: Regression with ARIMA Errors

1/5

- For the last 4 years, PH tractors is running an expensive marketing and farmer connect program to boost their sales.
- They are interested in learning the impact of this program on overall sales.
- As a data science consultant we are helping them with this effort.
- This is a problem that requires a thorough analysis followed by creative solutions and scientific monitoring mechanism.

Sales & Marketing: Regression with ARIMA Errors

2/5

- We will build models based on regression with ARIMA errors and compare them with the pure play ARIMA model.
- This analysis will provide some clues towards effectiveness of the marketing program.
- However, this analysis will not be conclusive for finding shortcomings and enhancements for the program which will require further analysis and creative solutions.

Sales & Marketing: Regression with ARIMA Errors

3/5

- Key Observations:
- This looks promising with quite a high correlation coefficient ($\rho > 0.8$).
- However, there is a danger in analyzing non-stationary time series data.
- Since two uncorrelated series can display high correlation because of time series trend in data.

Sales & Marketing: Regression with ARIMA Errors

4/5

- Key Observations:
 - In this case, PH is a growing company and the latent factor is 'growth' of the company.
 - Hence both its sales and marketing expenses can be on an upward curve independent of each other.

Sales & Marketing: Regression with ARIMA Errors

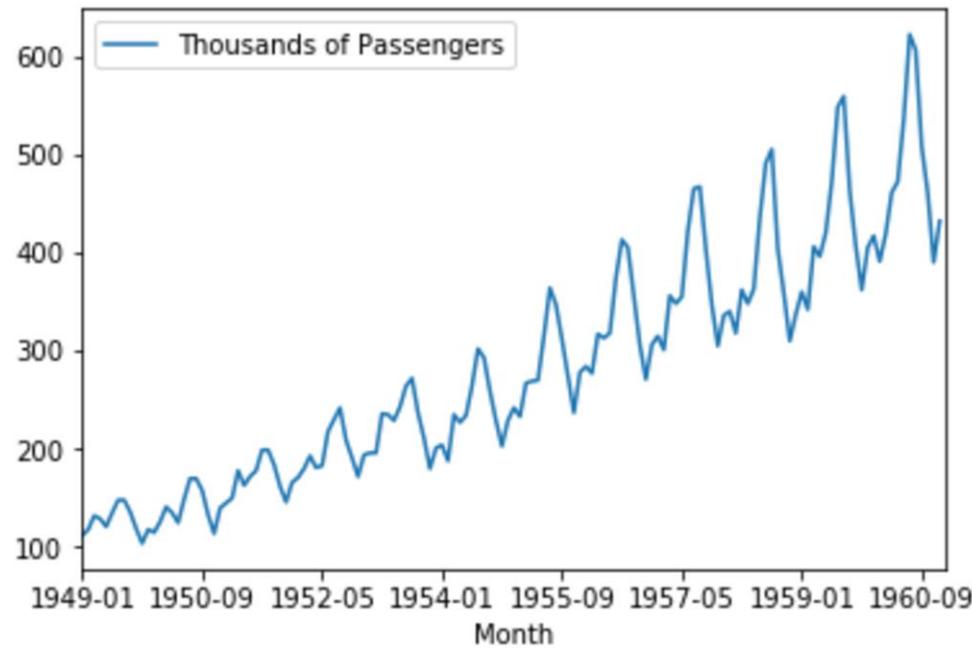
5/5

- Key Observations:
 - To investigate that a better way is to find the correlation between stationary data obtained through differencing of marketing expenditure and the tractor sales data individually.
 - The correlation plot for stationary data:

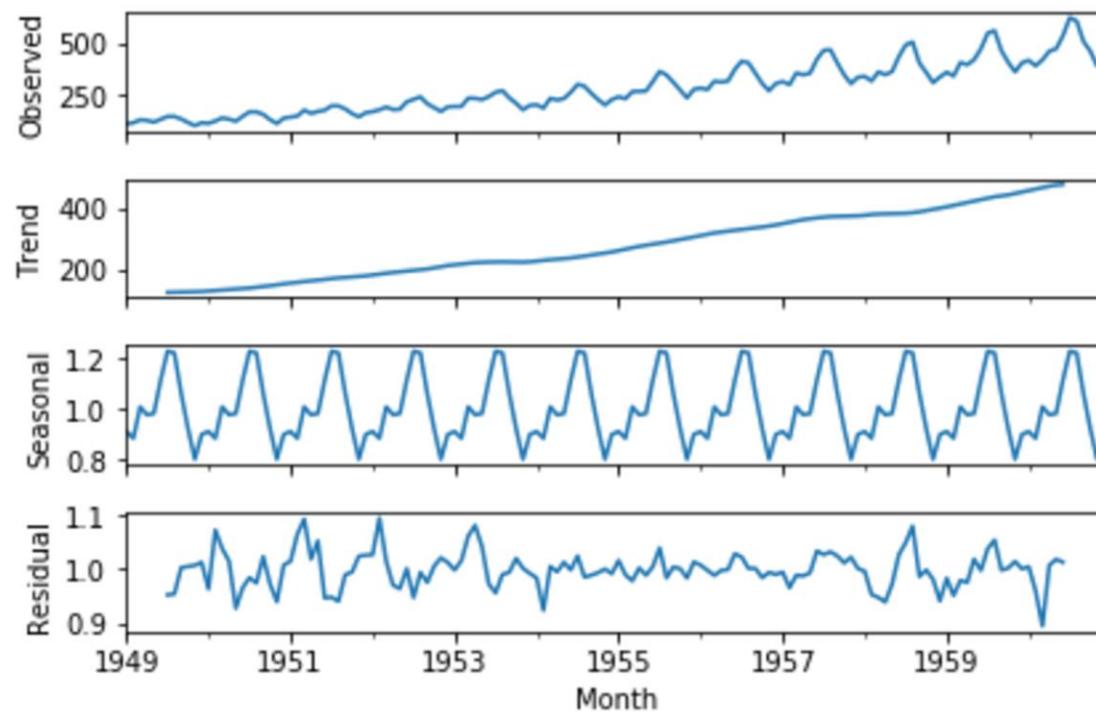
ETS Models

- ETS Models (Error-Trend-Seasonality)
 - Exponential Smoothing
 - Trend Methods Models
 - ETS Decomposition

- ETS Decomposition - Airline Passengers



- ETS Decomposition - Airline Passengers



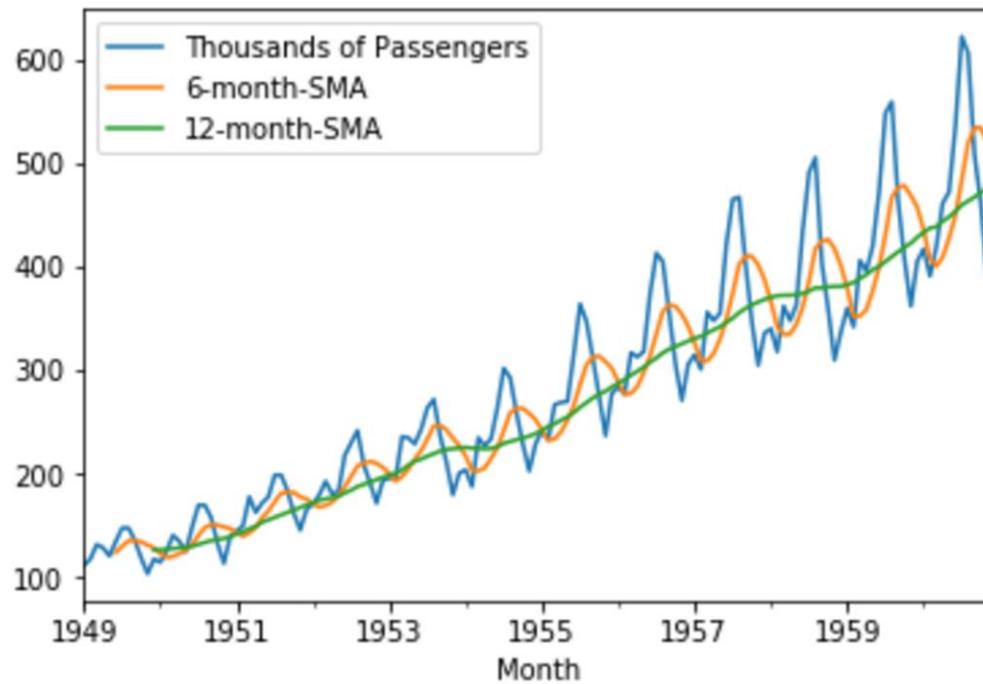
- Time Series Decomposition with ETS (Error-Trend-Seasonality).
- Visualizing the data based off its ETS is a good way to build an understanding of its behaviour.

- ETS (Error-Trend-Seasonality) Models will take each of those terms for “smoothing” and may add them, multiply them, or even just leave some of them out.
- Based off these key factors, we can try to create a model to fit our data.

- Time Series Decomposition with ETS (Error-Trend-Seasonality).
- Visualizing the data based off its ETS is a good way to build an understanding of its behaviour.

EWMA Models

- SMA - Simple Moving Averages



- EWMA- Exponentially Weighted Moving Averages
- Basic SMA has some "weaknesses".
 - Does not really inform you about possible future behaviour, all it really does is describe trends in your data.

- EWMA- Exponentially Weighted Moving Averages
- Basic SMA has some "weaknesses".
 - To help fix some of these issues, we can use an EWMA (Exponentially-weighted moving average).

- EWMA will allow us to reduce the lag effect from SMA and it will put more weight on values that occurred more recently (by applying more weight to the more recent values, thus the name).

ARIMA Models

- AutoRegressive Integrated Moving Average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model.

- Both of those models (ARIMA and ARMA) are fitted to time series data either to better understand the data or to predict future points in the series (forecasting).

- ARIMA (Autoregressive Integrated Moving Averages)
 - Non-seasonal ARIMA
 - Seasonal ARIMA

- ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.

- Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where parameters p, d, and q are non-negative integers.

- Parts of ARIMA model
- AR (p): Autoregression
 - A regression model that utilizes the dependent relationship between a current observation and observations over a previous period

- Parts of ARIMA model
- I (d): Integrated.
 - Differencing of observations (subtracting an observation from an observation at the previous time step) in order to make the time series stationary.

- Parts of ARIMA model
- MA (q): Moving Average.
 - A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

AutoCorrelation Plots

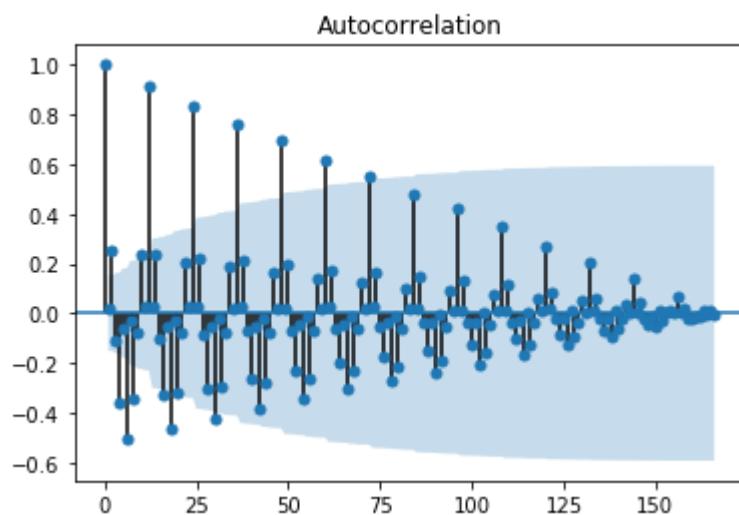
- An autocorrelation plot (also known as a Correlogram) shows the correlation of the series with itself, lagged by x time units.
- So the y axis is the correlation and the x axis is the number of time units of lag.

- Imagine taking your time series of length T , copying it, and deleting the first observation of copy 1 and the last observation of copy 2.

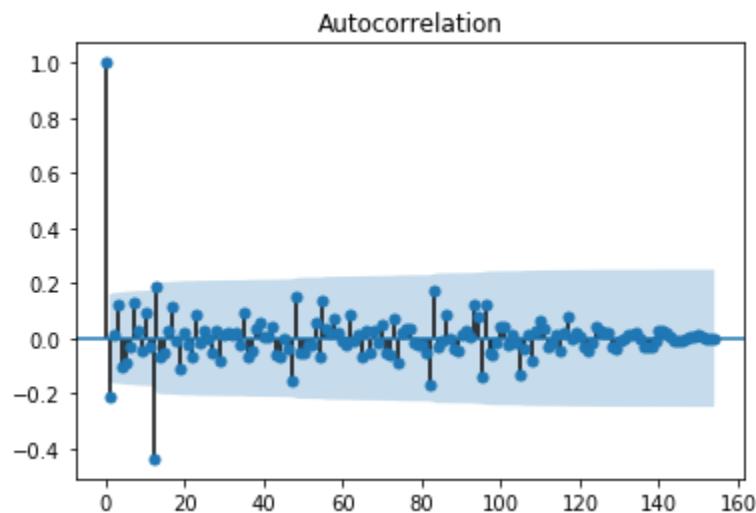
- Now you have two series of length $T-1$ for which you calculate a correlation coefficient.
- This is the value of the vertical axis at $x=1$ in your plots.

- It represents the correlation of the series lagged by one time unit.
- You go on and do this for all possible time lags x and this defines the plot.

- Gradual Decline



- Sharp Drop-off



- In general you would use either AR or MA, using both is less common.
- When actually applying the AR and MA terms, you will set values of p or q.

- If the autocorrelation plot shows positive autocorrelation at the first lag (lag-1), then it suggests to use the AR terms in relation to the lag

- If the autocorrelation plot shows negative autocorrelation at the first lag, then it suggests using MA terms.
- This will allow you to decide what actual values of p,d, and q to provide your ARIMA model.

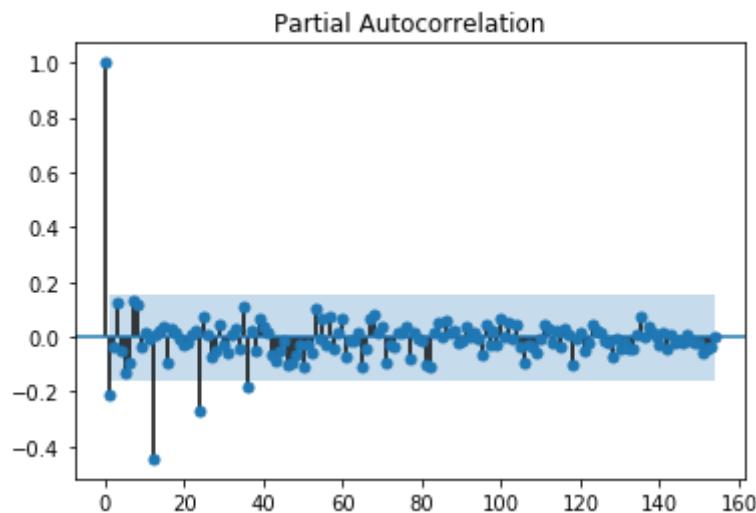
- p: The number of lag observations included in the model.
- d: The number of times that the raw observations are differenced
- q: The size of the moving average window, also called the order of moving average.

- There are also partial autocorrelation plots!

- In general, a partial correlation is a conditional correlation.
- It is the correlation between two variables under the assumption that we know and take into account the values of some other set of variables.

- For instance, consider a regression context in which y = response variable and x_1 , x_2 , and x_3 are predictor variables.
- The partial correlation between y and x_3 is the correlation between the variables determined taking into account how both y and x_3 are related to x_1 and x_2 .

- an example of what the plot can look like:



- Typically a sharp drop after lag "k" suggests an AR-k model should be used.
- If there is a gradual decline, it suggests an MA model.

- Identification of an AR model is often best done with the PACF.
- Identification of an MA model is often best done with the ACF rather than the PACF.

- Finally once you've analyzed your data using ACF and PACF you are ready to begin to apply ARIMA or Seasonal ARIMA, depending on your original data.
- We will provide the p,d, and q terms for the model.

- An ARIMA will then take three terms p,d, and q.
(We'll see this in the coding example)
- For seasonal ARIMA there will be an additional set of P,D,Q terms that we will see.

dplyr

a grammar of
data manipulation

Group	dose 1	dose 2
A	3	3
A	4	5
B	3	1
B	1	3
C	1	3
C	2	2

A 3 3

A 4 5

B 3 1

B 1 3

C 1 3

C 2 2

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

n	min	mean	max
6	4	5.2	9

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9

Group	Total
A	15

B	3	1	4
B	1	3	4

B	8
---	---

C	1	3	4
C	2	2	4

C	8
---	---

n	min	mean	max
6	4	5.2	9

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

Group	Sum
A	6
A	9
B	4
B	4
C	4
C	4

Group	Sum
A	6
C	4
C	4

tbl
%>%



database



American Airlines



American Eagle



FRONTIER
AIRLINES

jetBlue
AIRWAYS®

SkyWest
AIRLINES*

 **DELTA**



tbl

select
filter
arrange
mutate
summarize

Group	first dose	second dose	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

select

Group	Sum
A	6
A	9
B	4
B	4
C	4
C	4

select

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

filter

Group	dose 1	dose 2	Sum
A	3	3	6
C	1	3	4
C	2	2	4

filter

Group	dose 1	dose 2	Sum
C	1	3	4
A	3	3	6
A	4	5	9
B	3	1	4
B	2	3	4
C	5	2	4

arrange

Group	dose 1	dose 2	Sum
C	1	3	4
B	2	3	4
B	3	1	4
A	3	3	6
A	4	5	9
C	5	2	4

arrange

Group	dose 1	dose 2
A	3	3
A	4	5
B	3	1
B	1	3
C	1	3
C	2	2

mutate

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

mutate

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

summarise

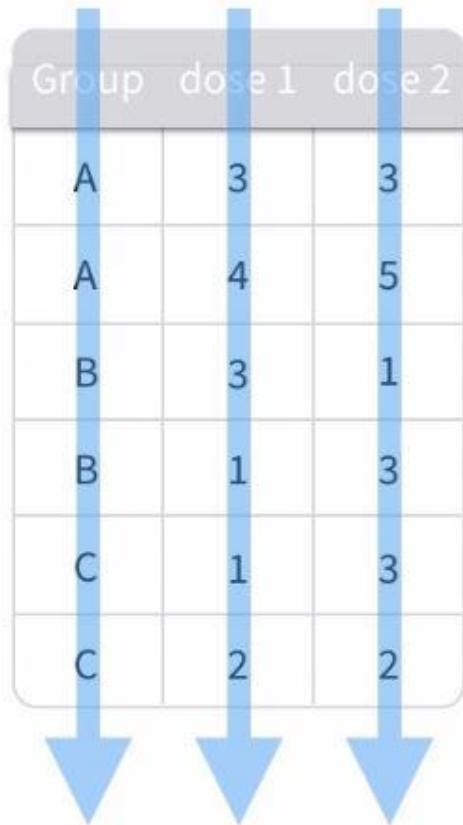
Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

n	min	mean	max
6	4	5.2	9

summarise

variables

Group	dose 1	dose 2
A	3	3
A	4	5
B	3	1
B	1	3
C	1	3
C	2	2



variables

observations

	Group	dose 1	dose 2	
	A	3	3	→
	A	4	5	→
	B	3	1	→
	B	1	3	→
	C	1	3	→
	C	2	2	→

select

mutate

filter

arrange

summarize

select

mutate

variables

filter

arrange

summarize

select

mutate

variables

filter

arrange

observations

summarize

select

mutate

variables

filter

arrange

observations

summarize

groups

Year	ActualElapsedTime
Month	AirTime
DayofMonth	ArrDelay
DayofWeek	DepDelay
DepTime	Origin
ArrTime	Dest
UniqueCarrier	Distance
FlightNum	TaxiIn
TailNum	TaxiOut
	Cancelled
	CancellationCode
	Diverted

Year	ActualElapsedTime
Month	AirTime
DayofMonth	ArrDelay
DayofWeek	DepDelay
DepTime	Origin
ArrTime	Dest
UniqueCarrier	Distance
FlightNum	TaxiIn
TailNum	TaxiOut
	Cancelled
	CancellationCode
	Diverted

tbl

columns to
select

```
select(df, Group, Sum)
```

tbl

columns to
select

select(df, Group, Sum)

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

tbl

columns to
select

select(df, Group, Sum)

Group	Sum
A	6
A	9
B	4
B	4
C	4
C	4

Group	dose 1	dose 2
A	3	3
A	4	5
B	3	1
B	1	3
C	1	3
C	2	2

mutate

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

mutate

length x width x height =

length	width	height
2	3	3
2	4	5
3	3	1
1	1	3

length x width x height = volume

length	width	height	volume
2	3	3	18
2	4	5	40
3	3	1	9
1	1	3	3

length x width x height = volume

mass / volume =

mass	volume
50	10
45	15
35	10

length x width x height = volume

mass / volume = density

mass	volume	density
50	10	5
45	15	3
35	10	3.5

```
mutate(h1, loss = ArrDelay - DepDelay)
```

tbl

```
mutate(h1, loss = ArrDelay - DepDelay)
```

tbl

new column
name

```
mutate(h1, loss = ArrDelay - DepDelay)
```

tbl

new column
name

=

expression

```
mutate(h1, loss = ArrDelay - DepDelay)
```

tbl

new column
name

=

expression

```
mutate(h1, loss = ArrDelay - DepDelay)
```

Year	ArrDelay	DepDelay	loss
2011	-10	0	-10
2011	-9	1	-10
2011	-8	-8	0
2011	3	3	0
2011	-3	5	-8
2011	-7	-1	-6

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

filter

Group	dose 1	dose 2	Sum
A	3	3	6
C	1	3	4
C	2	2	4

filter

tbl

logical test

```
filter(hflights, Cancelled == 1)
```

tbl

logical test

```
filter(hflights, Cancelled == 1)
```

Year	Cancelled	Dest
2011	0	DFW
2011	1	DFW
2011	0	ELP
2011	1	ELP

tbl

logical test

```
filter(hflights, Cancelled == 1)
```

Year	Cancelled	Dest
2011	1	DFW
2011	1	ELP

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	2	3	4
C	1	3	4
C	5	2	4

arrange

Group	dose 1	dose 2	Sum
C	1	3	4
B	2	3	4
B	3	1	4
A	3	3	6
A	4	5	9
C	5	2	4

arrange

tbl

column
name

```
arrange(hflights, DepDelay)
```

Year	DepDelay	Dest
2011	-2	DFW
2011	3	DFW
2011	0	ELP
2011	10	ELP

tbl

column
name

```
arrange(hflights, DepDelay)
```

Year	DepDelay	Dest
2011	-2	DFW
2011	0	ELP
2011	3	DFW
2011	10	ELP

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

summarise

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

n	min	mean	max
6	4	5.2	9

summarise

tbl

new column
name

=

expression

```
summarise(df, sum = sum(A),  
          avg = mean(B),  
          var = var(B))
```

tbl

new column
name

=

expression

```
summarise(df, sum = sum(A),  
          avg = mean(B),  
          var = var(B))
```

A	B	C
105	6	20
108	3	18
144	3	7
132	5	8

tbl

new column
name

=

expression

```
summarise(df, sum = sum(A),  
          avg = mean(B),  
          var = var(B))
```

A	B	C
105	6	20
108	3	18
144	3	7
132	5	8

sum avg var

489	4.25	44.92
-----	------	-------

A	B	C
105	6	20
108	3	18
144	3	7
132	5	8



summarise

sum	avg	var
489	4.25	44.92

```
a1 <- select(a, X, Y, Z)
a2 <- filter(a1, X > Y)
a3 <- mutate(a2, Q = X + Y + Z)
a4 <- summarise(a3, all = sum(Q))
```

```
a1 <- select(a, X, Y, Z)
a2 <- filter(a1, X > Y)
a3 <- mutate(a2, Q = X + Y + Z)
a4 <- summarise(a3, all = sum(Q))
```

```
summarise(  
  mutate(  
    filter(  
      select(a, X, Y, Z),  
      X > Y),  
      Q = X + Y + Z),  
    all = sum(Q))  
)
```

%>%

magrittr

some
object

pipe

some
function

arguments

```
object %>% function(____, arg2, arg3, ...)
```

```
summarise(  
  mutate(  
    filter(  
      select(a, X, Y, Z),  
      X > Y),  
      Q = X + Y + Z),  
  all = sum(Q))  
)
```

```
a %>%  
  select(X, Y, Z) %>%  
  filter(X > Y) %>%  
  mutate(Q = X + Y + Z) %>%  
  summarise(all = sum(Q))
```

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

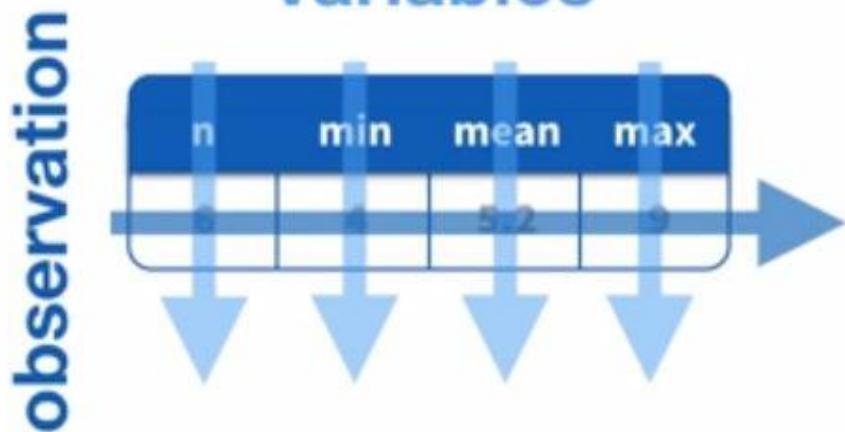


n	min	mean	max
6	4	5.2	9

summarise

Group	dose 1	dose 2	Sum
A	3	3	6
A	4	5	9
B	3	1	4
B	1	3	4
C	1	3	4
C	2	2	4

variables



Group	dose 1	dose 2	Sum	
A	3	3	6	
A	4	5	9	



Group	Total
A	15

B	3	1	4	
B	1	3	4	



B	8
---	---

C	1	3	4	
C	2	2	4	



C	8
---	---

n	min	mean	max
6	4	5.2	9

group_by

tbl

column to
group by

group_by(df, Group)

tbl

column to
group by

```
group_by(df, Group)
```

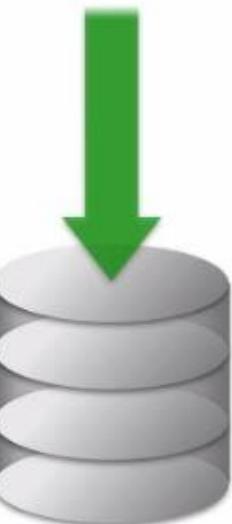
```
df %>%
```

```
  group_by(Group)
```

data frame

data table

database



database

WHAT IS TEXT MINING ?

- An exploration and analysis of textual (natural language) data to identify facts, relationships and assertions
- Process of analysing collections of textual materials in order to capture key concepts and themes and uncover hidden relationships and trends
- Examples:
 - ✓ Classical Spam Filtering
 - ✓ Review Analysis
 - ✓ Tweets Analysis

WHAT IS MISUNDERSTOOD AS TEXT MINING ?

- Information Retrieval
 - ✓ Information Retrieval is a domain that deals with most effective ways of retrieving information according to user needs and it is more concerned with search engine
- Examples: Search Engines like Google

WHAT IS MISUNDERSTOOD AS TEXT MINING ?

- Information Extraction
 - ✓ Information Extraction (IE) is about locating specific items in textual data
 - ✓ In information extraction we just extract the already known information which was not properly formatted
- Example: Home Address

WHAT IS MISUNDERSTOOD AS TEXT MINING ?

- Natural Language Processing (NLP)
 - ✓ NLP is based on latent features of the text and uses those in its methods and text mining is based on observed features
 - ✓ NLP considers the context in the text, while text mining does not

TEXT MINING SOURCES

- Comments (Social Networking Sites)
- Tweets
- Sales Reports
- Emails
- Blogs
- Word Documents

TYPES OF DOCUMENTS IN TEXT MINING

- Structured Documents
 - ✓ Survey Forms
 - ✓ Claims
- Semi Structured Documents
 - ✓ Job Listings
 - ✓ Invoices
- Unstructured (Free Format) Documents
 - ✓ Blogs

TEXT MINING PROCESS (STEPS)

- Given Data (Text)
- Text Preprocessing
- Feature Generation/Extraction
- Feature Selection
- Text Mining Methods
- Results Evaluation

TEXT PREPROCESSING

- The main idea is to extract (identify) the words from the texts by removing unnecessary data
- Words can be extracted by using following two techniques:
 - ✓ Lexical Analysis also known as Tokenization
 - ✓ Syntactical Analysis also known as part of speech tagging (using Brill POS Tagger)

TEXT PREPROCESSING

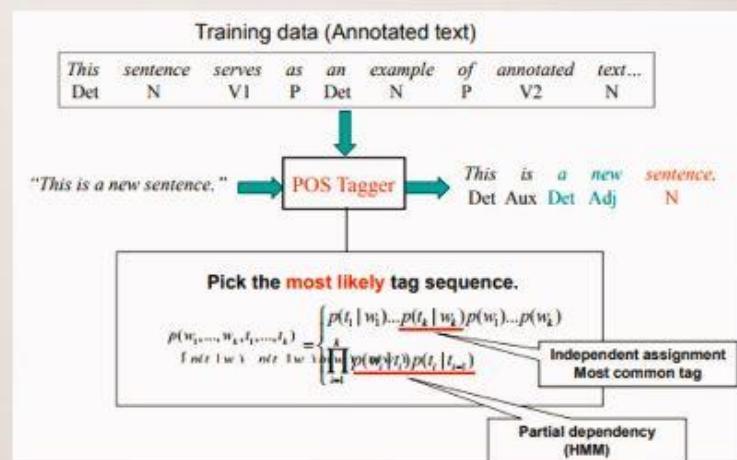
- Lexical Analysis:
 - ✓ Can be done using delimiters or by using a dictionary
- Easy to do, but sometimes not efficient
 - ✓ For example: swimming pool
- Lexical Analysis using Dictionary

TEXT PREPROCESSING

- Syntactic Analysis:

- ✓ Can be done using dictionary or by using a POS TAGGER

- Using a POS Tagger:



Taken from: ChengXiang Zhai

FEATURE EXTRACTION

- The main goal in this step is to extract good subset of words to represent the text documents in the collection
- Can be achieved by following methods:
 - ✓ Stop Word Elimination (remove uninformative words)
 - ✓ Stemming and Lemmatization

FEATURE EXTRACTION

- Stop Word Elimination:
- Can be done by using following two methods
 - ✓ Using a already maintained list of stop words
 - ✓ Bu using some statistical approach

FEATURE EXTRACTION

- Stemming:
 - ✓ To cut the end of words by some heuristic based process
- Examples:
 - ✓ Shoe and Shoes
 - ✓ Story and Stories

FEATURE EXTRACTION

- Lemmatization:
 - ✓ Another efficient way to reduce the vocabulary size and this is done by using a dictionary
- Examples:
 - ✓ Walking and Walk
 - ✓ Good and Better

WEIGHTING MODELS

- Main goal is to convert bag of word to vector space model

$$d_i = (w_{i1}, w_{i2}, \dots, w_{it}) \in \mathbf{R}^t$$

- w_{ij} is the weight of j^{th} term in document d_i , where j belongs to the index composed of t terms.
- Approaches:
 - ✓ Boolean Model
 - ✓ Term Frequency (TF)
 - ✓ Term Frequency Inverse Document Frequency (TFIDF)

TERM FREQUENCY MODEL

- This model takes into account the number of occurrences of words in the document

$$d_i = (w_{i1}, w_{i2}, \dots, w_{it}) \in \mathbf{R}^t$$

w_{ij} : the number of times jth term occurs in document d_i

$$w_{ij} = tf_{ij}$$

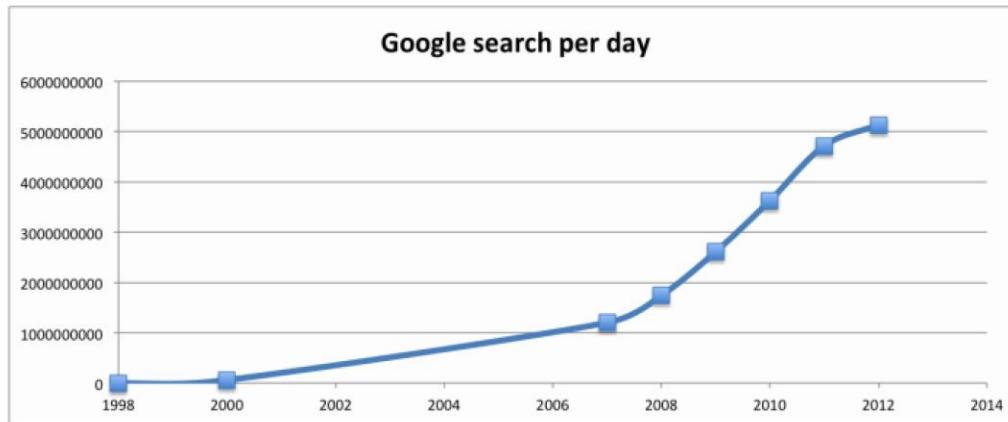
- Example:

- ✓ I need to practice, if I want to pass
- ✓ D=[2, 1, 2, 1, 1, 1, 1]

Tf-Idf and Cosine similarity

- In the year **1998** Google handled **9800** average search queries every day.
- In **2012** this number shot up to **5.13 billion** average searches per day.

The graph given below shows this astronomical growth.



Diploma in Data Science & Big Data Analytics

Naresh IT Opposite Satyam Theatre, Ameerpet, Hyderabad 040-2374666, 23734842

Google Search- How it works?

Let's consider 3 documents to show how this works. Take some time to go through them.

- **Document 1:** The game of life is a game of everlasting learning
- **Document 2:** The unexamined life is not worth living
- **Document 3:** Never stop learning

Let us imagine that you are doing a search on these documents with the following query:

"life learning"

The query is a **free text query**.

It means a query in which the terms of the query are typed freeform into the search interface, without any connecting search operators.

Diploma in Data Science & Big Data Analytics

Naresh IT Opposite Satyam Theatre, Ameerpet, Hyderabad 040-2374666, 23734842

Step 1: Term Frequency (TF)

Term Frequency also known as TF measures the number of times a term (word) occurs in a document.

Given below are the terms and their frequency on each of the document.

TF for Document 1

Document1	the	game	of	life	is	a	everlasting	learning
Term Frequency	1	2	2	1	1	1	1	1

TF for Document 2

Document2	the	unexamined	life	is	not	worth	living
Term Frequency	1	1	1	1	1	1	1

TF for Document 3

Document3	never	stop	learning
Term Frequency	1	1	1

Step 1 (continued): Normalized Term Frequency (TF)

- In reality each document will be of different size.
- On a large document the frequency of the terms will be much higher than the smaller ones.
- Hence we need to normalize the document based on its size.
- A simple trick is to divide the term frequency by the total number of terms.

e.g:

In Document 1 the term game occurs two times.

The total number of terms in the document is 10.

Hence the normalized term frequency is $2 / 10 = 0.2$.

Diploma in Data Science & Big Data Analytics

Naresh IT Opposite Satyam Theatre, Ameerpet, Hyderabad 040-2374666, 23734842

Step 1 (continued): Normalized Term Frequency (

Given below are the normalized term frequency for all the documents.

Normalized TF for Document 1

Document1	the	game	of	life	is	a	everlasting	learning
Normalized TF	0.1	0.2	0.2	0.1	0.1	0.1	0.1	0.1

Normalized TF for Document 2

Document2	the	unexamined	life	is	not	worth	living
Normalized TF	0.142857	0.142857	0.142857	0.142857	0.142857	0.142857	0.142857

Normalized TF for Document 3

Document3	never	stop	learning
Normalized TF	0.333333	0.333333	0.333333

Step 2: Inverse Document Frequency (IDF)

The main purpose of doing a search is to find out relevant documents matching the query. In the first step all terms are considered equally important.

- Certain terms that **occur too frequently have little power** in determining the relevance.
- Solution: **Weigh down the effects of too frequently occurring terms.**
- The terms that **occur less in the document can be more relevant.**
- Solution: **Weigh up the effects of less frequently occurring terms.**

Logarithms helps us to solve this problem.

Given below is the IDF for terms occurring in all the documents. Since the terms: the, life, is, learning occurs in 2 out of 3 documents they have a lower score compared to the other terms that appear in only one document.

Step 2 (continued): Inverse Document Frequency (IDF)

Computing IDF for the term **game**:

$IDF(\text{game}) = 1 + \log_e(\text{Total Number Of Documents} / \text{Number Of Documents with term } \text{game} \text{ in it})$

There are 3 documents in all = Document1, Document2, Document3

The term game appears in Document1

$$\begin{aligned} IDF(\text{game}) &= 1 + \log_e(3 / 1) \\ &= 1 + 1.098726209 \\ &= 2.098726209 \end{aligned}$$

Step 2 (continued): Inverse Document Frequency (IDF)

Terms	IDF
the	1.405507153
game	2.098726209
of	2.098726209
life	1.405507153
is	1.405507153
a	2.098726209
everlasting	2.098726209
learning	1.405507153
unexamined	2.098726209
not	2.098726209
worth	2.098726209
living	2.098726209
never	2.098726209
stop	2.098726209

Diploma in Data Science & Big Data Analytics

Naresh IT Opposite Satyam Theatre, Ameerpet, Hyderabad 040-2374666, 23734842

Step 3: TF * IDF

Remember we are trying to find out relevant documents for the query: **life learning**

- For each term in the query multiply its normalized term frequency with its IDF on each document.



- Given below is TF * IDF calculations for **life** and **learning** in all the documents.

	Document1	Document2	Document3
life	0.140550715	0.200786736	0
learning	0.140550715	0	0.468502384

Step 4 (Continued): Vector Space Model – Cosine Simi

- From each document we derive a vector.
- The set of documents in a collection then is viewed as a set of vectors in a vector space. Each term will have its own axis.
- Using the formula given below we can find out the similarity between any two documents.

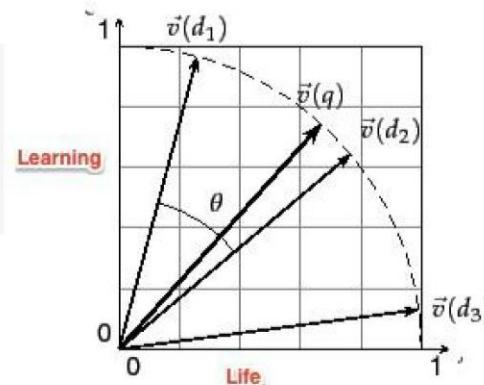
Cosine Similarity (d_1, d_2) = Dot product(d_1, d_2) / $\|d_1\| * \|d_2\|$

Dot product (d_1, d_2) = $d_1[0] * d_2[0] + d_1[1] * d_2[1] * \dots * d_1[n] * d_2[n]$

$\|d_1\| = \text{square root}(d_1[0]^2 + d_1[1]^2 + \dots + d_1[n]^2)$

$\|d_2\| = \text{square root}(d_2[0]^2 + d_2[1]^2 + \dots + d_2[n]^2)$

- Vectors deals only with numbers. In this example we are dealing with text documents.
- We used **TF and IDF** to convert text into numbers so that it can be represented by a vector.



Diploma in Data Science & Big Data Analytics

Naresh IT Opposite Satyam Theatre, Ameerpet, Hyderabad 040-2374666, 23734842

Step 4 (Continued): Vector Space Model – Cosine Similarity

The query entered by the user can also be represented as a vector.

We will calculate the TF*IDF for the query

	TF	IDF	TF*IDF
life	0.5	1.405507153	0.702753576
learning	0.5	1.405507153	0.702753576

Note:

The cosine value is always between -1 and 1 : the cosine of a small angle is near 1 , and the cosine of a large angle near 180 degrees is close to -1 . This is good, because small angles should map to high similarity, near 1 , and large angles should map to near -1 .

Step 4 (Continued): Vector Space Model – Cosine Similarity

Let us now calculate the cosine similarity of the query and Document1.

```
Cosine Similarity(Query, Document1) = Dot product(Query, Document1) / ||Query|| * ||Document1||

Dot product(Query, Document1)
= ((0.702753576) * (0.140550715) + (0.702753576)*(0.140550715))
= 0.197545035151

||Query|| = sqrt((0.702753576)2 + (0.702753576)2) = 0.993843638185

||Document1|| = sqrt((0.140550715)2 + (0.140550715)2) = 0.198768727354

Cosine Similarity(Query, Document1) = 0.197545035151 / (0.993843638185) * (0.198768727354)
= 0.197545035151 / 0.197545035151
= 1
```

Given below is the similarity scores for all the documents and the query

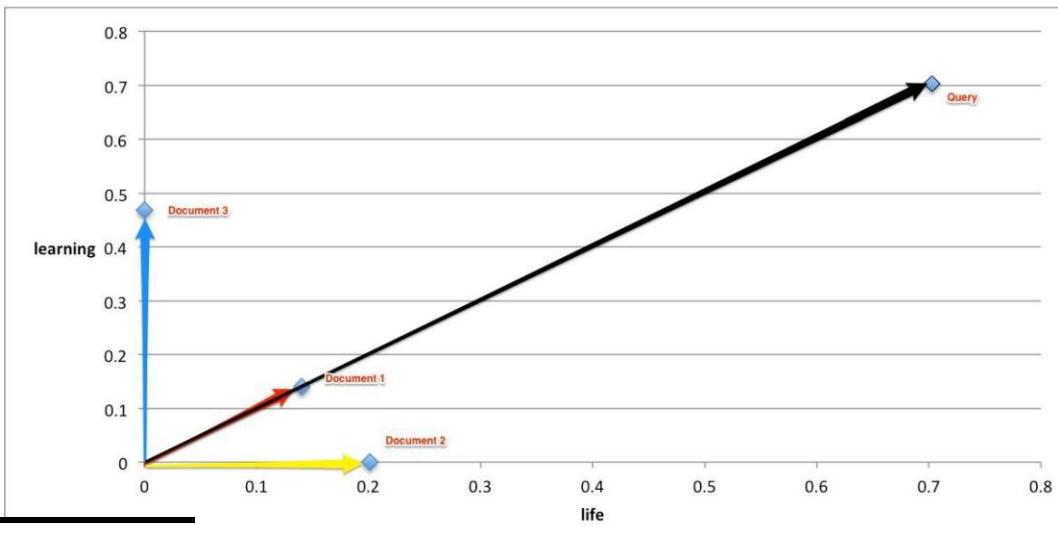
	Document1	Document2	Document3
Cosine Similarity	1	0.707106781	0.707106781

Diploma in Data Science & Big Data Analytics

Naresh IT Opposite Satyam Theatre, Ameerpet, Hyderabad 040-2374666, 23734842

Step 4 (Continued): Vector Space Model – Cosine Sim

- ✓ Below is the plot of vector values for the query and documents in 2-dimensional space of life and learning.
- ✓ Document1 has the highest score of 1.
- ✓ This is not surprising as it has both the terms **life** and **learning**.



DIMENSION REDUCTION

- The main goal in this step is to reduce the size of vocabulary to avoid overfitting (curse of dimensionality)
- Can be achieved by following methods:
 - ✓ Document Frequency Thresholding
 - ✓ χ^2 statistic
 - ✓ PCA (Principal Component Analysis)
 - ✓ Latent semantic Analysis (LSA)/LSI

TEXT MINING APPLICATIONS/TASKS

- Keyword Based Association Analysis:

“In keyword based association analysis, we collect a set of keywords that occur frequently together and then we try to find out the relationship”

- Examples:
 - ✓ Document Tagging

TEXT MINING APPLICATIONS/TASKS

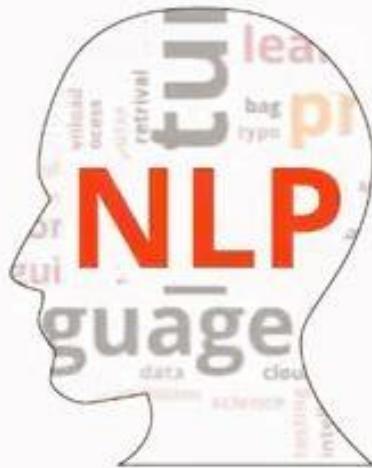
- Text mining as classification:

“We try to automatically classify large collection of documents in predefined categories”

- Classification can be accomplished by various techniques:
 - ✓ K neighbor KNN
 - ✓ Decision Trees
 - ✓ Neural Networks (Deep Learning)
 - ✓ SVM (Support Vector Machines)
 - ✓ Naïve Bayes Classifier (Probabilistic Model)

Natural Language Processing (NLP)

Natural Language Processing is an automated way to understand and analyze natural human languages and extract information from such data by applying machine algorithms.



Machine learning
algorithms

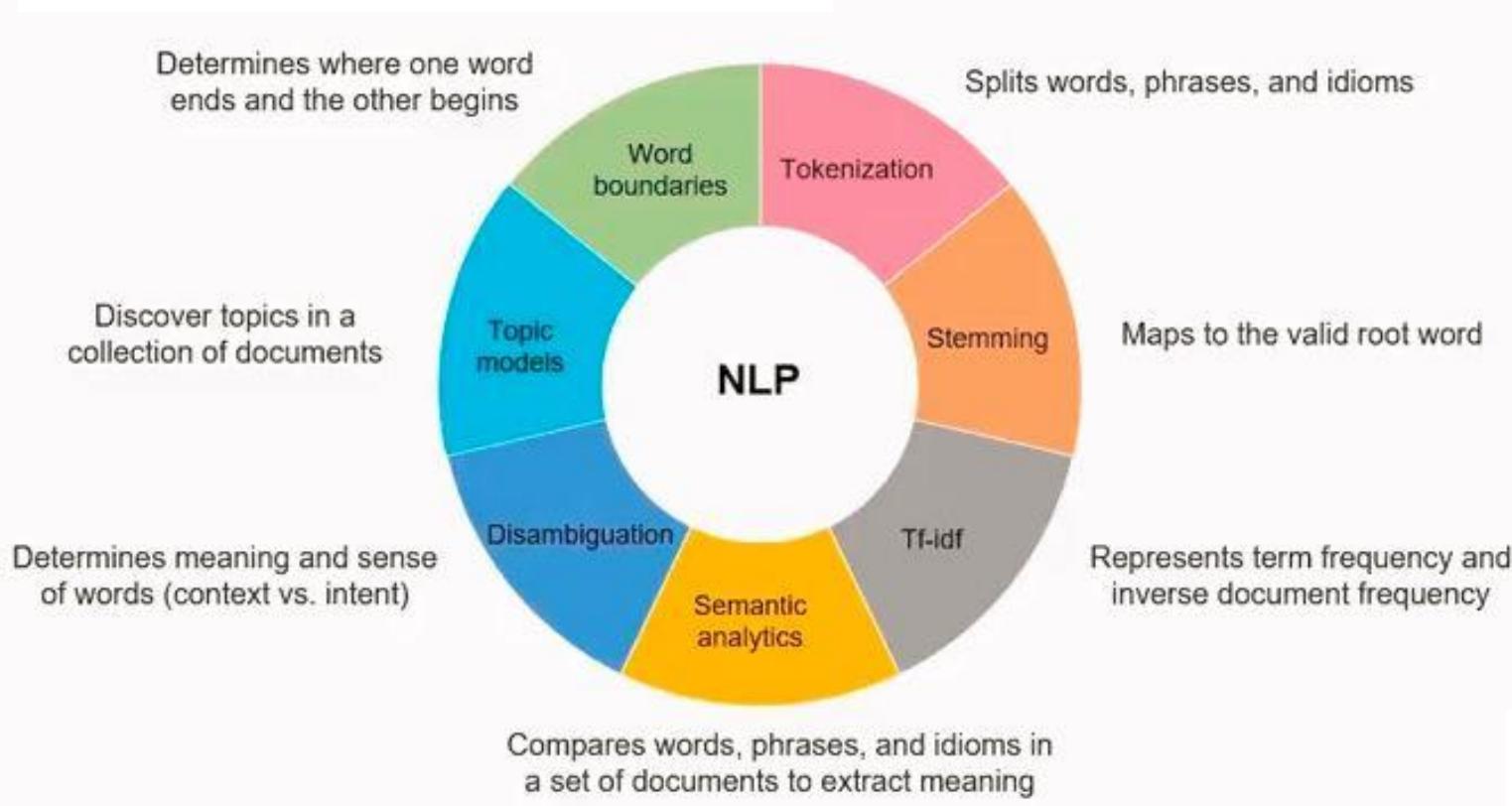
Why Natural Language Processing

- Analyzing tons of data
- Identifying various languages and dialects
- Applying quantitative analysis
- Handling ambiguities

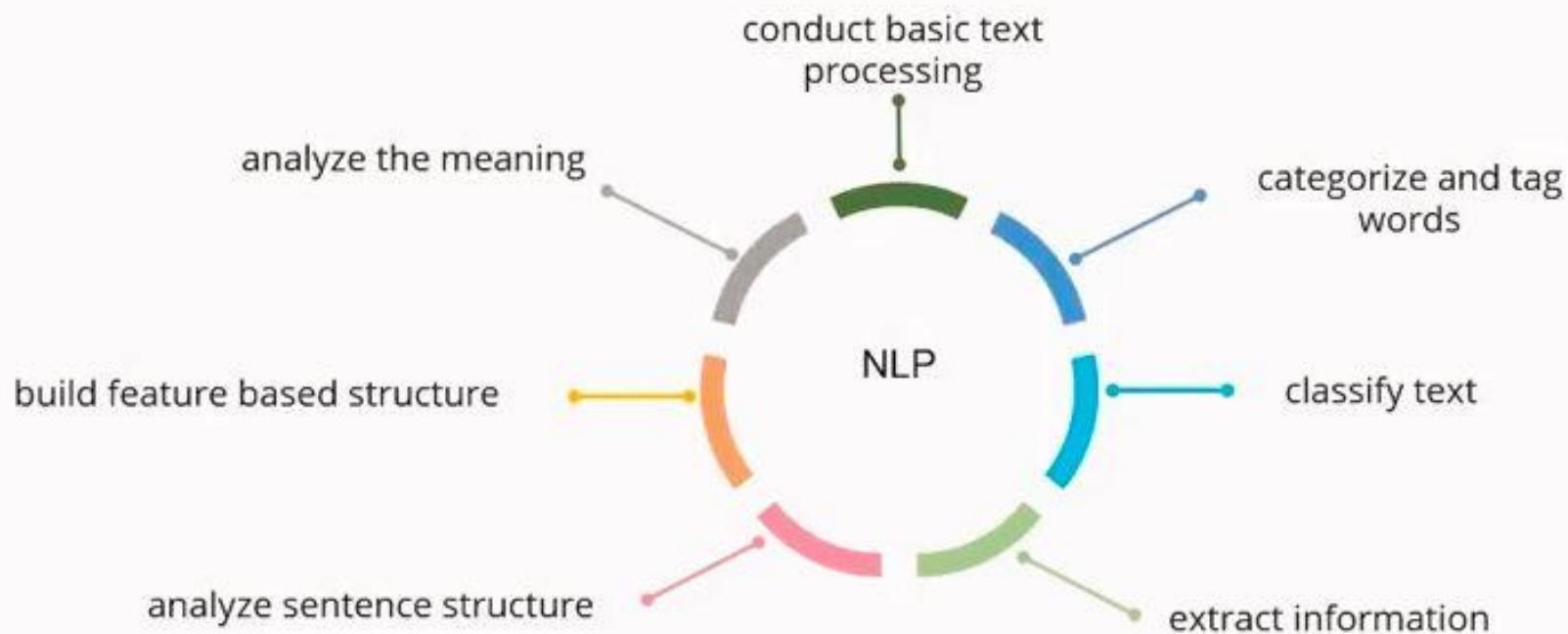


NLP Terminology

NLP terminologies:

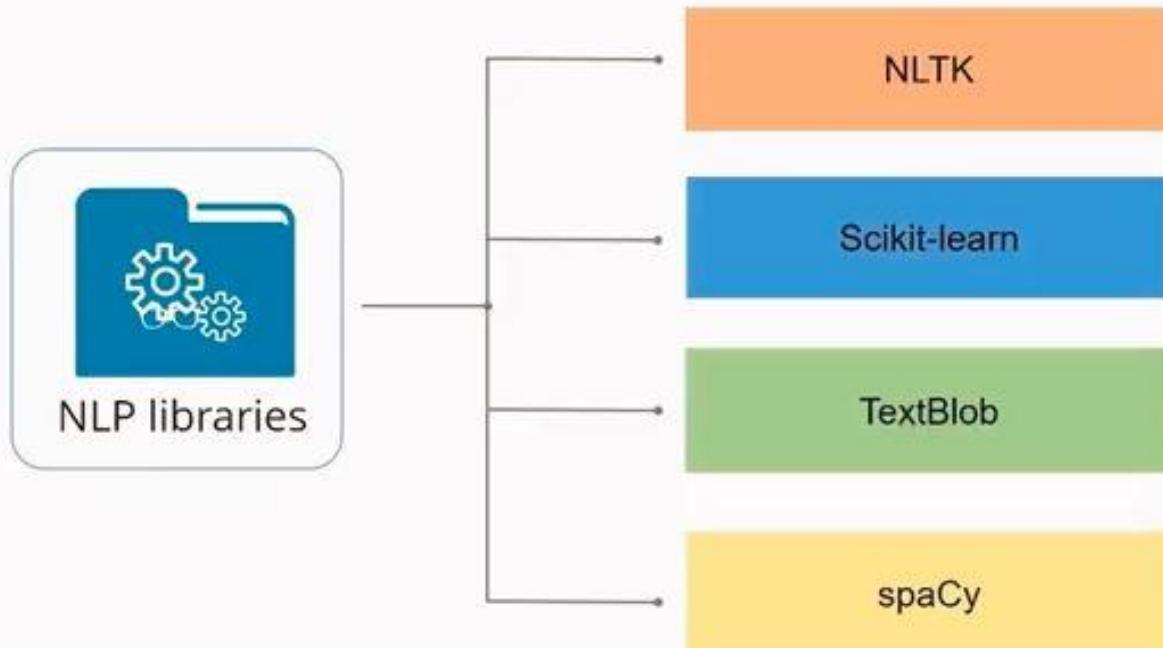


The NLP Approach for Text Data

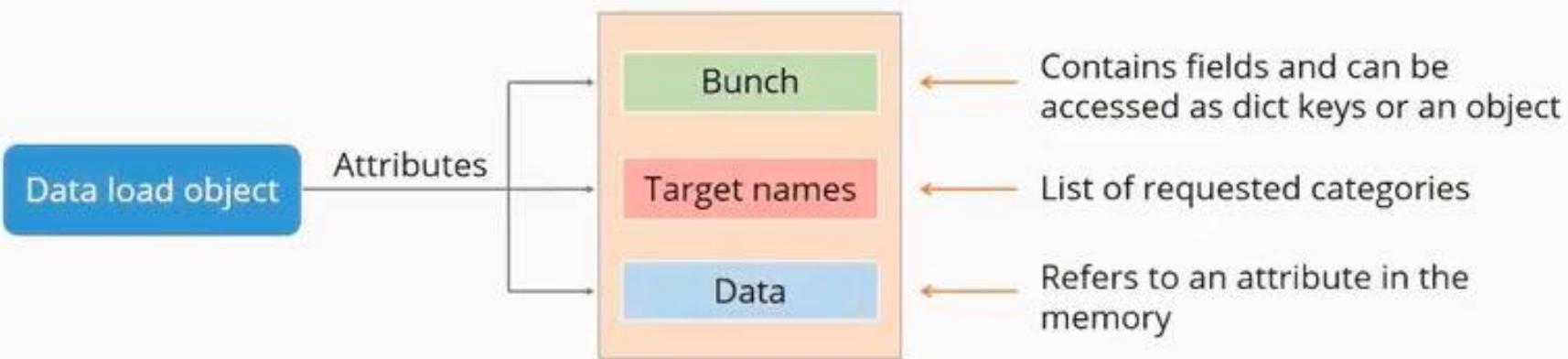


Major NLP Libraries

The major NLP libraries used in Python are:



Modules to Load Content and Category



Feature Extraction

Feature extraction is a technique to convert the content into the numerical vectors to perform machine learning.



Text feature extraction

For example: Large datasets or documents

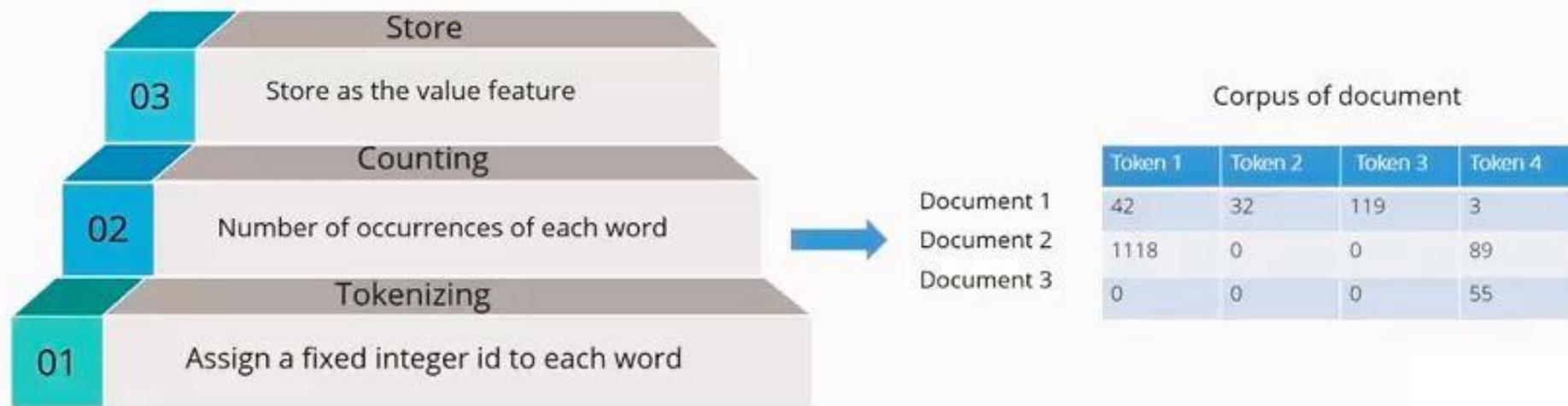


Image feature extraction

For example: Patch extraction,
hierarchical clustering

Bag of Words

Bag of words is used to convert text data into numerical feature vectors with a fixed size.



Model Training

An important task in model training is to identify the right model for the given dataset. The choice of model completely depends on the type of dataset.



Supervised

Models predict the outcome of new observations and datasets, and classify documents based on the features and response of a given dataset

Example: Naïve Bayes, SVM, linear regression, K-NN neighbors



Unsupervised

Models identify patterns in the data and extract its structure. They are also used to group documents using clustering algorithms.

Example: K-means

Naïve Bayes Classifier

Advantages

- It is efficient as it uses limited CPU and memory.
- It is fast as the model training takes less time.

Uses

- Naïve Bayes is used for sentiment analysis, email spam detection, categorization of documents, and language detection.
- Multinomial Naïve Bayes is used when multiple occurrences of the words matter.

Bayes' theorem (too) simplified

- Probability of an event A = $P(A)$ is between 0 and 1
- Bayes' theorem gives the conditional probability of an event A given event B has already occurred.

$$P(A/B) = P(A \text{ intersect } B) * P(A) / P(B)$$

- Example

- There are 100 patients
- Probability of a patient having diabetes is $P(A) = .2$
- Probability of patient having diabetes (A) given that the patient's age is > 50 (B) is $P(A/B) = .4$

Naïve Bayes Classification

- Application of Bayes' theorem to ML
- The target variable becomes event A
- The predictors become events B₁ – B_n
- We try to find P(A / B₁-B_n)

Age	BMI	Is Diabetic	
24	22	N	Probability of Is Diabetic = Y given that Age = 24 and BMI = 22
41	36	Y	Probability of Is Diabetic – Y given that Age = 41 and BMI = 36

Model building and prediction

- The model generated stores the conditional probability of the target for every possible value of the predictor.

Salary	Overall	Age						Gender	
		1 to 20	20 to 30	30 to 40	40 to 50	50 to 60	60 to 100	Female	Male
< 50K	.75	0.1	0.3	0.25	0.17	0.1	0.08	0.39	0.61
> 50K	.25	0.03	0.08	0.3	0.32	0.2	0.07	0.15	0.85
Overall		.08	.24	.26	.21	.12	.08	.33	.67

- When a new prediction needs to be done, the conditional probabilities are applied using Bayes' formula to find the probability
 - To predict for Age = 25
 - $P(\text{Salary} < 50K / \text{Age}=25) = 0.3 * 0.75 / 0.24 = \sim 0.92$
 - $P(\text{Salary} > 50K / \text{Age}=25) = 0.08 * 0.25 / 0.24 = \sim 0.08$

Summary – Naïve Bayes

Advantages

- Simple and fast
- Works well with noisy and missing data
- Provides probabilities of the result
- Very good with categorical data

Shortcomings

- Limited Accuracy
- Assumes predictors are independent
- Not good with large numeric features

Used in

- Medical diagnosis
- Spam filtering
- Document classification

Text Summarization

A NLP based approach

Text Summarization -Techniques

- A simple Natural Language Processing based approach
- A Deep NLP based approach

Text Summarization – Sample paragraph

Thank you all so very much. Thank you to the Academy. Thank you to all of you in this room. I have to congratulate the other incredible nominees this year. *The Revenant* was the product of the tireless efforts of an unbelievable cast and crew.

Text Summarization – Tokenization

1. Thank you all so very much.
2. Thank you to the Academy.
3. Thank you to all of you in this room.
4. I have to congratulate the other incredible nominees this year.
5. *The Revenant* was the product of the tireless efforts of an unbelievable cast and crew.

Text Summarization – Preprocess

1. thank you all so very much
2. thank you to the academy
3. thank you to all of you in this room
4. i have to congratulate the other incredible nominees this year
5. *the revenant* was the product of the tireless efforts of an unbelievable cast and crew

Text Summarization – Histogram

Word	Count	Word	Count	Word	Count
thank	3	in	1	revenant	1
you	4	this	2	was	1
all	2	room	1	product	1
so	1	i	1	tireless	1
very	1	have	1	efforts	1
much	1	congratulate	1	an	1
to	3	other	1	unbelievable	1
the	5 ✓	incredible	1	cast	1
academy	1.	nominees	1	and	1
of	3	year	1	crew	3

Text Summarization – Weighted Histogram

Word	Weight	Word	Weight	Word	Weight
thank	3/5	in	1/5	revenant	1/5
you	4/5	this	2/5	was	1/5
all	2/5	room	1/5	product	1/5
so	1/5	i	1/5	tireless	1/5
very	1/5	have	1/5	efforts	1/5
much	1/5	congratulate	1/5	an	1/5
to	3/5	other	1/5	unbelievable	1/5
the	5/5	incredible	1/5	cast	1/5
academy	1/5	nominees	1/5	and	1/5
of	3/5	year	1/5	crew	1/5

Maximum is
5

Text Summarization – Weighted Histogram

Word	Weight	Word	Weight	Word	Weight
thank	0.5	in	0.2	revenant	0.2
you	0.8	this	0.4	was	0.2
all	0.4	room	0.2	product	0.2
so	0.2	i	0.2	tireless	0.2
very	0.2	have	0.2	efforts	0.2
much	0.2	congratulate	0.2	an	0.2
to	0.6	other	0.2	unbelievable	0.2
the	1	incredible	0.2	cast	0.2
academy	0.2	nominees	0.2	and	0.2
of	0.6	year	0.2	crew	0.2

Text Summarization – Sentence Scores

thank	0.5
you	0.8
all	0.4
so	0.2
very	0.2
much	0.2
	2.3

Text Summarization – Sentence Scores

Sentence	Score
thank you all so very much	2.3
thank you to the academy	3.1
thank you to all of you in this room	4.3
i have to congratulate the other incredible nominees this year	3.4
<i>the revenant</i> was the product of the tireless efforts of an unbelievable cast and crew	6.2

Text Summarization – Sort by Score

Sentence	Score
<i>the revenant</i> was the product of the tireless efforts of an unbelievable cast and crew .	6.2
thank you to all of you in this room	4.3
i have to congratulate the other incredible nominees this year	3.4
thank you to the academy	3.1
thank you all so very much	2.3

Text Summarization – Pickle N Largest

Sentence	Score
<i>the revenant</i> was the product of the tireless efforts of an unbelievable cast and crew	6.2
thank you to all of you in this room	4.3

Text Summarization – Summary done

The Revenant was the product of the tireless efforts of an unbelievable cast and crew.

Thank you to all of you in this room.

Word2Vec

Introduction to the Word2Vec

BOW, TFIDF - Problems

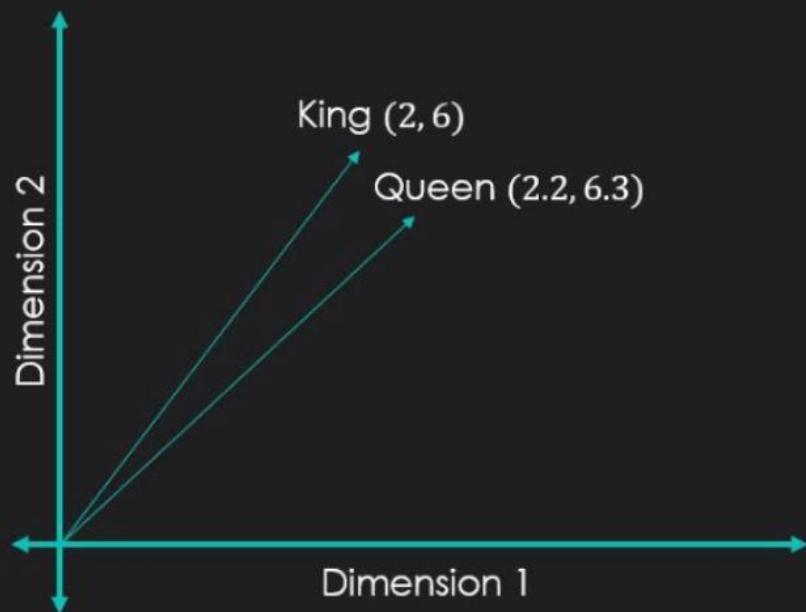
- Semantic information of the words is not stored. Even in TF-IDF model we only give more importance to the uncommon words.
- There's a chance of overfitting the model. Overfitting a scenario when model performs very well with your dataset but fails miserably when applied to any new dataset.

Word2Vec – The solution

- In this model, each word is represented as vector of 32 or more dimension instead of a single number.
- Relation between different words is preserved.

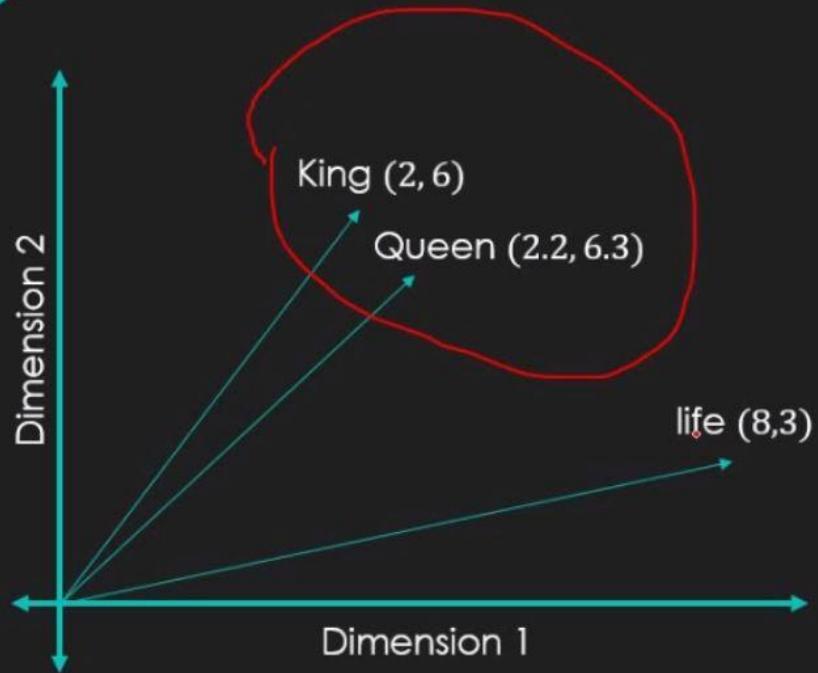
Word2Vec – Graphical Representation

2 dimensional



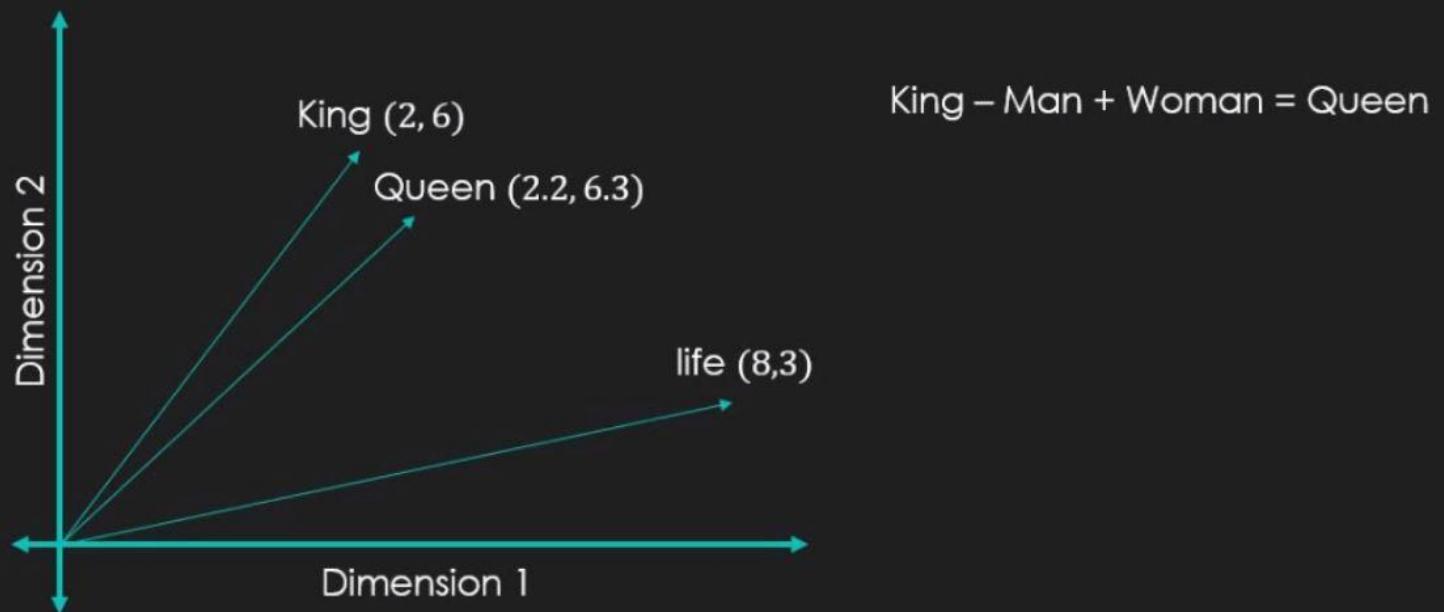
Word2Vec – Graphical Representation

2 dimensional



Word2Vec – Graphical Representation

2 dimensional



Word2Vec – Extracting sentence meaning

“Sachin Tendulkar is the Roger Federer of Cricket”

Word2Vec – Extracting sentence meaning

“Sachin Tendulkar is the Roger Federer of Cricket”

Roger Federer – tennis + cricket = Sachin Tendulkar



Word2Vec – Steps to build the model

- Scrape through a **huge** dataset like the whole Wikipedia.
- Create a matrix with all the unique words in the dataset.
The matrix represents the occurrence relation between
the words.
- Split the matrix into two thin matrices.
- We have the model.

Word2Vec – Sample Dataset

going
to
today
i
am
it
is
rain
not
outside

“it is going to rain today”
“today i am not going outside”

“i am going to watch the season premiere”

Word2Vec – Steps to build the model

Word2Vec – Word Matrix Formation

Words	going	to	today	i	am	it	is	rain	not	outside
going	3	2	2	2	2	1	1	1	1	1
to	2	2	1	1	1	1	1	1	0	0
today	2	1	2	1	1	1	1	1	1	1
i	2	1	1	2	2	0	0	0	1	1
am	2	1	1	2	2	0	0	0	1	1
it	1	1	1	0	0	1	1	1	0	0
is	1	1	1	0	0	1	1	1	0	0
rain	1	1	1	0	0	1	1	1	0	0
not	1	0	1	1	1	0	0	0	1	1
outside	1	0	1	1	1	0	0	0	1	1

Word2Vec – Splitting into smaller matrices

Words	Dimension 1	Dimension 2
going		
to		
today		
i		
am		
it		
is		
rain		
not		
outside		

Word2Vec – Splitting into smaller matrices

Words	Dimension 1	Dimension 2
going		
to		
today		
i		
am		
it		
is		
rain		
not		
outside		

Words	going	to	today	i	am	it	is	rain	not	outside
Dimension 1										
Dimension 2										

$$A * A^T$$

•

Word2Vec – Word Vectors

Words	Dimension 1	Dimension 2
going		
to		
today		
i		
am		
it		
is		
rain		
not		
outside		

Word2Vec – Word Vectors

Words	Dimension 1	Dimension 2
going	$X_{1\text{going}}$	$X_{2\text{going}}$
to	$X_{1\text{to}}$	$X_{2\text{to}}$
today	$X_{1\text{today}}$	$X_{2\text{today}}$
i	X_{1i}	X_{2i}
am	$X_{1\text{am}}$	$X_{2\text{am}}$
it	$X_{1\text{it}}$	$X_{2\text{it}}$
is	$X_{1\text{is}}$	$X_{2\text{is}}$
rain	$X_{1\text{rain}}$	$X_{2\text{rain}}$
not	$X_{1\text{not}}$	$X_{2\text{not}}$
outside	$X_{1\text{outside}}$	$X_{2\text{outside}}$

Word2Vec – Word Vectors

going = ($X_{1\text{going}}$, $X_{2\text{going}}$, ..., $X_{300\text{going}}$)