References:

<https://medium.com/acing-ai/salesforce-ai-interview-questions-acing-the-ai-interview-75e177c4734>

<https://medium.com/acing-ai/microsoft-ai-interview-questions-acing-the-ai-interview-be6972f790ea>

<https://medium.com/acing-ai/apple-ai-interview-questions-acing-the-ai-interview-803a65b0e795>

<https://medium.com/acing-ai/amazon-ai-interview-questions-acing-the-ai-interview-3ed4e671920f>

<https://medium.com/acing-ai/uber-ai-interview-questions-acing-the-ai-interview-9532794bc057>

<https://medium.com/acing-ai/steps-to-ace-the-ai-interview-part-1-298249080e59>

<https://medium.com/acing-ai/steps-to-ace-the-ai-interview-part-2-b25f91582f5f>

<https://medium.com/acing-ai/google-ai-interview-questions-acing-the-ai-interview-1791ad7dc3ae>

<https://medium.com/acing-ai/facebook-ai-interview-questions-acing-the-ai-interview-5982add0af55>

<https://www.analyticsvidhya.com/blog/2016/09/40-interview-questions-asked-at-startups-in-machine-learning-data-science/>

<https://www.analyticsvidhya.com/blog/2017/09/30-questions-test-k-nearest-neighbors-algorithm/>

<https://www.analyticsvidhya.com/blog/2017/08/skilltest-logistic-regression/>

<https://www.listendata.com/2017/03/predictive-modeling-interview-questions.html>

<https://www.analyticsvidhya.com/blog/2017/07/30-questions-to-test-a-data-scientist-on-linear-regression/>

<https://www.analyticsvidhya.com/blog/2016/12/45-questions-to-test-a-data-scientist-on-regression-skill-test-regression-solution/>

<https://medium.com/acing-ai/adobe-ai-interview-questions-acing-the-ai-interview-ef7a8099110b>

<https://www.listendata.com/2018/03/regression-analysis.html>

<https://www.analyticsvidhya.com/blog/2017/10/svm-skilltest/>

<https://vitalflux.com/decision-tree-algorithm-concepts-interview-question>

<https://www.analyticsvidhya.com/blog/2017/09/30-questions-test-tree-based-models/>

<https://www.analyticsvidhya.com/blog/2017/01/must-know-questions-deep-learning/>

<https://www.kdnuggets.com/tag/interview-questions>

LocalDate currentDate = LocalDate.now();

Age age3 = (birthDate) -> {

LocalDate currentDate = LocalDate.now();

if (birthDate != null)

return Period.between(birthDate, currentDate).getYears();

else

return 0;

};

//sort by price in descending order using anonymous inner class

Collections.sort(products, new Comparator<Product>() {

@Override

public int compare(Product o1, Product o2) {

return o2.getItemCost().compareTo(o1.getItemCost());

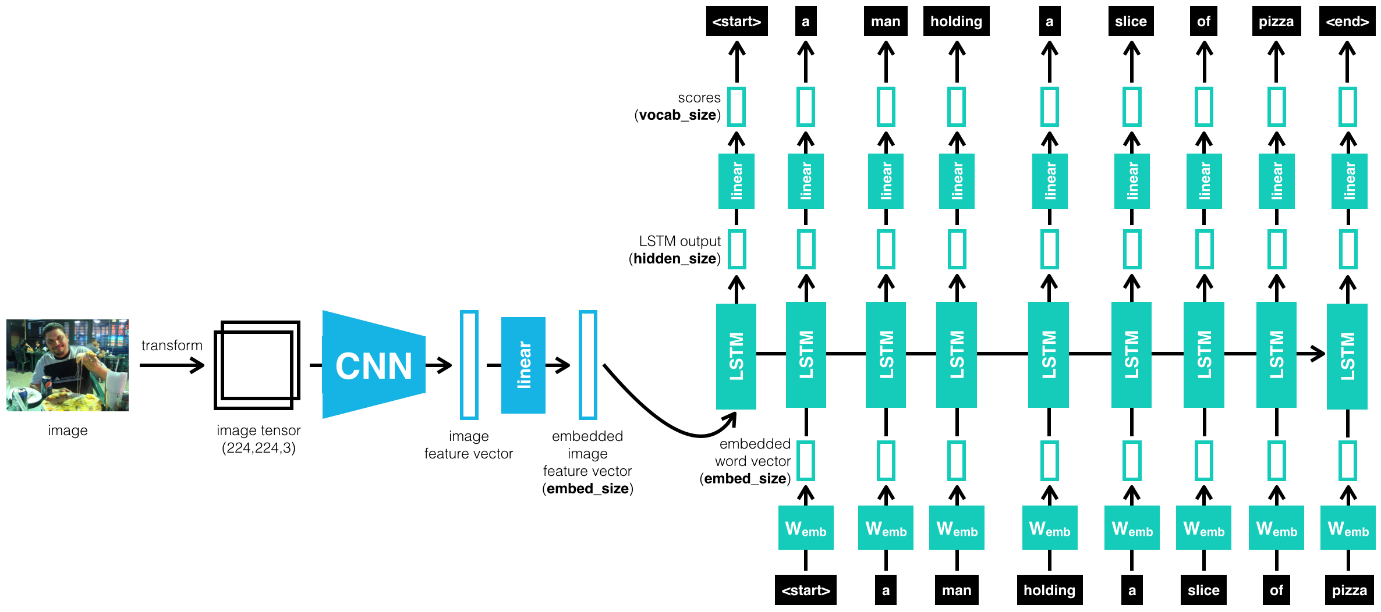
}

});

<https://www.qualtrics.com/uk/experience-management/research/sampling-methods/>

<https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>

Image captioning



|  |
| --- |
| import torch |
|  | import torch.nn as nn |
|  | import torchvision.models as models |
|  |  |
|  |  |
|  | class EncoderCNN(nn.Module): |
|  | def \_\_init\_\_(self, embed\_size): |
|  | super(EncoderCNN, self).\_\_init\_\_() |
|  | resnet = models.resnet50(pretrained=True) |
|  | for param in resnet.parameters(): |
|  | param.requires\_grad\_(False) |
|  |  |
|  | modules = list(resnet.children())[:-1] |
|  | self.resnet = nn.Sequential(\*modules) |
|  | self.embed = nn.Linear(resnet.fc.in\_features, embed\_size) |
|  |  |
|  | def forward(self, images): |
|  | features = self.resnet(images) |
|  | features = features.view(features.size(0), -1) |
|  | features = self.embed(features) |
|  | return features |
|  |  |
|  |  |
|  | class DecoderRNN(nn.Module): |
|  | def \_\_init\_\_(self, embed\_size, hidden\_size, vocab\_size, num\_layers=1): |
|  | super(DecoderRNN, self).\_\_init\_\_() |
|  | self.embedding = nn.Embedding(vocab\_size, embed\_size) |
|  | self.lstm = nn.LSTM(embed\_size, hidden\_size, num\_layers, batch\_first = True) |
|  | self.fc = nn.Linear(hidden\_size, vocab\_size) |
|  |  |
|  | def forward(self, features, captions): |
|  | captions = captions[:, :-1] |
|  | embed = self.embedding(captions) |
|  | embed = torch.cat((features.unsqueeze(1), embed), dim = 1) |
|  | lstm\_outputs, \_ = self.lstm(embed) |
|  | output = self.fc(lstm\_outputs) |
|  | return output |
|  |  |
|  | def sample(self, inputs, states=None, max\_len=20): |
|  | " accepts pre-processed image tensor (inputs) and returns predicted sentence (list of tensor ids of length max\_len) " |
|  | tensor\_tokens = [] |
|  | for i in range(max\_len): |
|  | out, states = self.lstm(inputs, states) |
|  | out = self.fc(out.squeeze(1)) |
|  | \_, predicted = out.max(1) |
|  | tensor\_tokens.append(predicted.item()) |
|  | inputs = self.embedding(predicted) |
|  | inputs = inputs.unsqueeze(1) |
|  | return tensor\_tokens |

A lot of researchers are using dropout with Computer Vision (CV) because they have a very big input size and almost never have enough data, so overfitting is the usual problem. And dropout is a regularization technique to prevent overfitting. A downside of dropout is that the cost function J is not well defined and it will be hard to debug (plot J by iteration). To solve that you'll need to turn off dropout, set all the keep\_prob s to 1, and then run the code and check that it monotonically decreases J and then turn on the dropouts again.

Andrew prefers to use L2 regularization instead of early stopping because this technique simultaneously tries to minimize the cost function and not to overfit which contradicts the orthogonalization approach (will be discussed further)

Gradient checking approximates the gradients and is very helpful for finding the errors in your backpropagation implementation but it's slower than gradient descent (so use only for debugging).

Don't use the gradient checking algorithm at training time because it's very slow. Use gradient checking only for debugging. If algorithm fails grad check, look at components to try to identify the bug. Don't forget to add lamda/(2m) \* sum(W[l]) to J if you are using L1 or L2 regularization. Gradient checking doesn't work with dropout because J is not consistent. You can first turn off dropout (set keep\_prob = 1.0 ), run gradient checking and then turn on dropout again. Run gradient checking at random initialization and train the network for a while maybe there's a bug which can be seen when w's and b's become larger (further from 0) and can't be seen on the first iteration (when w's and b's are very small).

Different initializations lead to different results Random initialization is used to break symmetry and make sure different hidden units can learn different things

Implications of L2-regularization on: cost computation: A regularization term is added to the cost backpropagation function: There are extra terms in the gradients with respect to weight matrices weights: weights end up smaller ("weight decay") - are pushed to smaller values.

The code inside an epoch should be vectorized. Mini-batch gradient descent works much faster in the large datasets.

Mini-batch size is a hyperparameter .

V(t) = beta \* v(t-1) + (1-beta) \* theta(t)

If we plot this it will represent averages over ~ (1 / (1 - beta)) entries: beta = 0.9 will average last 10 entries beta = 0.98 will average last 50 entries beta = 0.5 will average last 2 entrie

The momentum algorithm almost always works faster than standard gradient descent. The simple idea is to calculate the exponentially weighted averages for your gradients and then update your weights with the new values.

vdW = 0, vdb = 0 on iteration t: # can be mini-batch or batch gradient descent compute dw, db on current mini-batch vdW = beta \* vdW + (1 - beta) \* dW vdb = beta \* vdb + (1 - beta) \* db W = W - learning\_rate \* vdW b = b - learning\_rate \* vdb

decay\_rate is another hyperparameter . One technique equations is learning\_rate = (1 / (1 + decay\_rate \* epoch\_num)) \* learning\_rate\_0

Hyperparameters importance are (as for Andrew Ng): i. Learning rate. ii. Momentum beta. iii. Mini-batch size. iv. No. of hidden units. v. No. of layers. vi. Learning rate decay. vii. Regularization lambda. viii. Activation functions. ix. Adam beta1 & beta2 .

Batch Normalization speeds up learning. Why does Batch normalization work? The first reason is the same reason as why we normalize X. The second reason is that batch normalization reduces the problem of input values changing (shifting). Batch normalization does some regularization: Each mini batch is scaled by the mean/variance computed of that mini-batch. This adds some noise to the values Z[l] within that mini batch. So similar to dropout it adds some noise to each hidden layer's activations. This has a slight regularization effect. Using bigger size of the mini-batch you are reducing noise and therefore regularization effect. Don't rely on batch normalization as a regularization. It's intended for normalization of hidden units, activations and therefore speeding up learning. For regularization use other regularization techniques (L2 or dropout)

Almost all TensorFlow programs use this: with tf.Session() as session: # better for cleaning up in case of error/exception session.run(init) session.run(w)

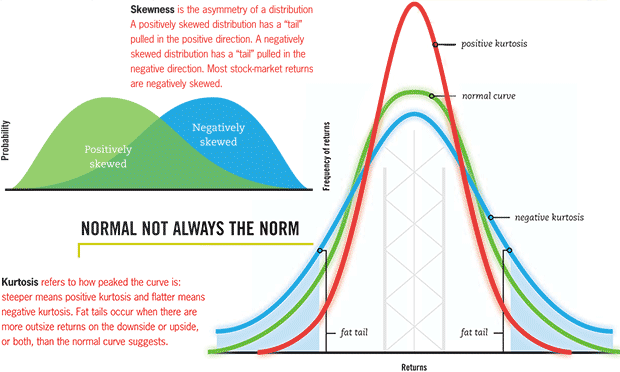
You have a lot of ideas for how to improve the accuracy of your deep learning system: Collect more data. Collect more diverse training set. Train algorithm longer with gradient descent. Try different optimization algorithm (e.g. Adam). Try bigger network. Try smaller network. Try dropout. Add L2 regularization. Change network architecture (activation functions, # of hidden units, etc.

An old way of splitting the data was 70% training, 30% test or 60% training, 20% dev, 20% test. In the modern deep learning if you have a million or more examples a reasonable split would be 98% training, 1% dev, 1% test.

If avoidable bias is large you have these options: Train bigger model. Train longer/better optimization algorithm (like Momentum, RMSprop, Adam). Find better NN architecture/hyperparameters search. iv. If variance is large you have these options: Get more training data. Regularization (L2, Dropout, data augmentation). Find better NN architecture/hyperparameters search.

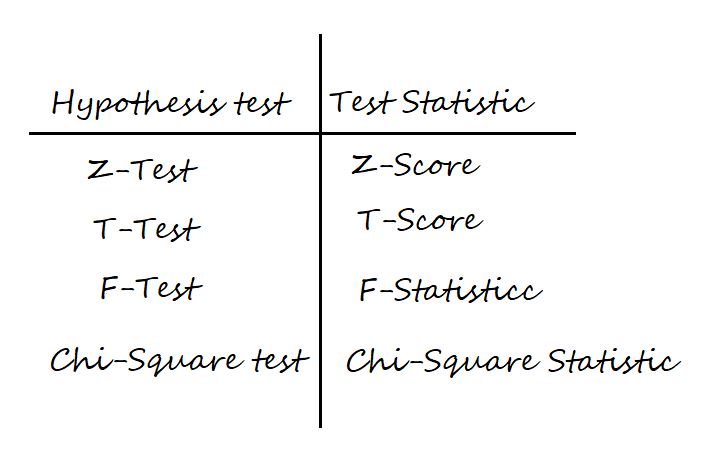
The steps you take to make your deep learning project: Setup dev/test set and metric Build initial system quickly Use Bias/Variance analysis & Error analysis to prioritize next steps.

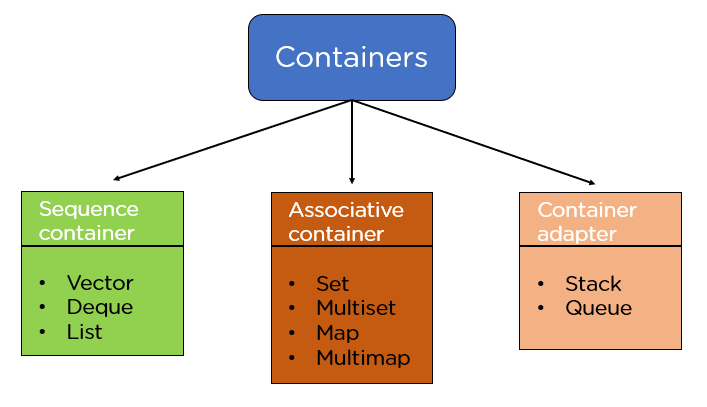
Some systems have multiple stages to implement. An end-to-end deep learning system implements all these stages with a single NN. Less hand-designing of components needed.



<https://www.analyticsvidhya.com/blog/2021/07/hypothesis-testing-made-easy-for-the-data-science-beginners/>

<https://towardsdatascience.com/everything-you-need-to-know-about-hypothesis-testing-part-i-4de9abebbc8a>





Optimization

Enumerative search 🡪 Random search 🡪 Hill climbing

Linear programming 🡪 Binary linear programming (branch and bound)

Simulated annealing Late-acceptance hill-climbing

Genetic algorithms , PSO , CMA

How to escape local minima

Allow larger jumps in the nbr function 2 Restart (iterated hill-climbing) 3 Allow disimproving moves (simulated annealing and late-acceptance hill-climbing) 4 Use multiple search points with information transfer between them (genetic algorithm)

Facial Keypoint detector

### Question 1: What optimization and loss functions did you choose and why?

**Answer**: Used Adam optimizer as it combines best properties of AdaGrad and RMSProp and adjust its parameters really well wrt time/epoch. Also I used Smooth L1 loss as loss function as its a really good regression loss (smooth loss function,prevents exploding gradients)<https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html>

### Question 2: What kind of network architecture did you start with and how did it change as you tried different architectures? Did you decide to add more convolutional layers or any layers to avoid overfitting the data?

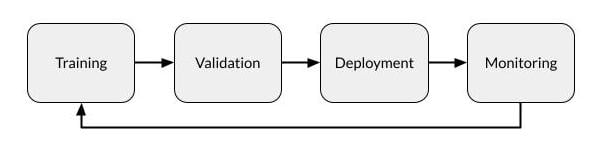
**Answer**: I increased number of convolution layers based on researrch paper : <https://arxiv.org/pdf/1710.00977.pdf> which is performing better as compared to other smaller models.Also used dropout on each layer to reduced overfitting. Total there are 5 cov layer and 4 fully connected layer, which felt good for this complex problem.

### Question 3: How did you decide on the number of epochs and batch\_size to train your model?

**Answer**: I increased bactch size to 20 (as I have 6gb GPU , it was possible to increase batch size) and number of epoch to 10 (This model can be trained for just 5 epoch and still perform well, but i trained for few epoch to reduce loss from 0.006 to 0.001)

<https://github.com/prakhargurawa/Facial-Keypoint-Detector/blob/main/models.py>

Machine learning deployment



Every deployed model has the potential to degrade over time due to such issues as:

* **Variance in deployed data.** Often, the data given to the model in deployment is not cleaned in the same manner as the training and testing data were, resulting in changes in model deployment.
* **Changes in data integrity.**Over weeks, months or years, changes in data being fed to the model can adversely affect model performance, such as changes in formats, renamed fields or new categories.
* **Data drift.**Changes in demographics, market shifts and so on can cause drift over time, making the training data less relevant to the current situation and the model’s results therefore less precise.
* **Concept drift.** Changes in the end users’ expectations of what constitutes a correct prediction can change over time, making the model’s predictions less relevant.

Learning points

Word2vec

Word2vec provides an option to choose between CBOW (continuous Bag of words) and skim-gram. Such parameters are provided during training of the model. One can have the option of using negative sampling or hierarchical softmax layer.

CBOW is several times faster than skip gram and provides a better frequency for frequent words whereas skip gram needs a small amount of training data and represents even rare words or phrases.

### What other options are available other than Hierarchical Softmax?

If speaking in a general sense for word embedding options available are Differentiated Softmax, CNN-Softmax, Importance Sampling, Adaptive Importance sampling, Noise Contrastive Estimations, Negative Sampling, Self-Normalization, and infrequent Normalization.

# [BERT Explained](https://www.lyrn.ai/2018/11/07/explained-bert-state-of-the-art-language-model-for-nlp/): State of the art language model for NLP

BERT (Bidirectional Encoder Representations from Transformers)

BERT’s key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling.

 In the paper, the researchers detail a novel technique named Masked LM (MLM) which allows bidirectional training in models in which it was previously impossible.

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT’s goal is to generate a language model, only the encoder mechanism is necessary.

## Masked LM (MLM)

Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence.

A picture containing graphical user interface

Description automatically generated

BERT can be used for a wide variety of language tasks, while only adding a small layer to the core model:

1. Classification tasks such as sentiment analysis are done similarly to Next Sentence classification, by adding a classification layer on top of the Transformer output for the [CLS] token.
2. In Question Answering tasks (e.g. SQuAD v1.1), the software receives a question regarding a text sequence and is required to mark the answer in the sequence. Using BERT, a Q&A model can be trained by learning two extra vectors that mark the beginning and the end of the answer.
3. In Named Entity Recognition (NER), the software receives a text sequence and is required to mark the various types of entities (Person, Organization, Date, etc) that appear in the text. Using BERT, a NER model can be trained by feeding the output vector of each token into a classification layer that predicts the NER label.

Shape

Description automatically generated with medium confidence

* **Precision**: Of all positive predictions, how many are really positive?
* **Recall**: Of all real positive cases, how many are predicted positive?

Precision measures the extent of error caused by False Positives (FPs) whereas recall measures the extent of error caused by False Negatives (FNs).

Increasing precision will decrease recall, and vice versa.

**Simply put, a z-score (also called a standard score) gives you an idea of how far from the**[**mean**](https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-median-mode/)**a data point is.** But more technically it’s a measure of how many [standard deviations](https://www.statisticshowto.com/probability-and-statistics/standard-deviation/) below or above the [population mean](https://www.statisticshowto.com/population-mean/)a [raw score](https://www.statisticshowto.com/raw-score/) is.

The **basic z score formula** for a [sample](https://www.statisticshowto.com/sample/)is:

**z = (x – μ) / σ**

For example, let’s say you have a test score of 190. The test has a mean (μ) of 150 and a [standard deviation](https://www.statisticshowto.com/probability-and-statistics/standard-deviation/) (σ) of 25. Assuming a [normal distribution](https://www.statisticshowto.com/probability-and-statistics/normal-distributions/), your z score would be:

* z = (x – μ) / σ
* = (190 – 150) / 25 = 1.6.
* A p-value is a statistical measurement used to validate a hypothesis against observed data.
* A p-value measures the probability of obtaining the observed results, assuming that the null hypothesis is true.
* The lower the p-value, the greater the statistical significance of the observed difference.
* A p-value of 0.05 or lower is generally considered statistically significant.

Text

Description automatically generated

Text, letter

Description automatically generated

A decorator is a design pattern in Python that allows a user to add new functionality to an existing object without modifying its structure. Decorators are usually called before the definition of a function you want to decorate.

### First Class Objects In Python, functions are [****first class objects****](https://www.geeksforgeeks.org/first-class-functions-python/) which means that functions in Python can be used or passed as arguments. ****Properties of first class functions:****

As stated above the decorators are used to modify the behaviour of function or class. In Decorators, functions are taken as the argument into another function and then called inside the wrapper function.

# Latent Semantic Analysis: intuition, math, implementation

TL;DR — Text data suffers heavily from high-dimensionality. Latent Semantic Analysis (LSA) is a popular, dimensionality-reduction techniques that follows the same method as Singular Value Decomposition.

LSA is one of the most popular Natural Language Processing (NLP) techniques for trying to determine themes within text mathematically. LSA is an unsupervised learning technique that rests on two pillars:

Note that LSA is an unsupervised learning technique — there is no ground truth. A whiteboard with writing on it

Description automatically generated with medium confidence

Deciding if a function is linear or not is of course not a matter of opinion or debate; there is a very simple [definition](https://en.wikipedia.org/wiki/Linear_function) of a linear function, which is roughly:

f(a\*x + b\*y) = a\*f(x) + b\*f(y)

for every x & y in the function domain and a & b constants.

The requirement "for every" means that, if we are able to find even a single example where the above condition does not hold, then the function is nonlinear.

Assuming for simplicity that a = b = 1, let's try x=-5, y=1 with f being the ReLU function:

f(-5 + 1) = f(-4) = 0

f(-5) + f(1) = 0 + 1 = 1

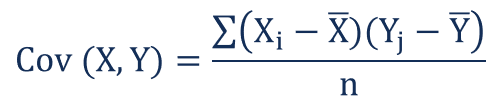
Graphical user interface, text, application, Excel

Description automatically generated

In mathematics and [statistics](https://corporatefinanceinstitute.com/resources/knowledge/basic-statistics-concepts/), covariance is a measure of the relationship between two random variables. The metric evaluates how much – to what extent – the variables change together. In other words, it is essentially a measure of the variance between two variables. However, the metric does not assess the dependency between variables.

Unlike the correlation coefficient, covariance is measured in units. The units are computed by multiplying the units of the two variables. The variance can take any positive or negative values. The values are interpreted as follows:

* **Positive covariance**: Indicates that two variables tend to move in the same direction.
* **Negative covariance**: Reveals that two variables tend to move in inverse directions.



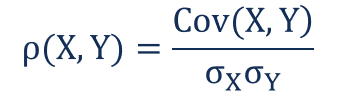
### Covariance vs. Correlation

Covariance and correlation both primarily assess the relationship between variables. The closest analogy to the relationship between them is the relationship between the variance and [standard deviation](https://corporatefinanceinstitute.com/resources/knowledge/standard-deviation/).

**Covariance** measures the total variation of two random variables from their expected values. Using covariance, we can only gauge the direction of the relationship (whether the variables tend to move in tandem or show an inverse relationship). However, it does not indicate the strength of the relationship, nor the dependency between the variables.

On the other hand, **correlation** measures the strength of the relationship between variables. Correlation is the scaled measure of covariance. It is dimensionless. In other words, the correlation coefficient is always a pure value and not measured in any units.

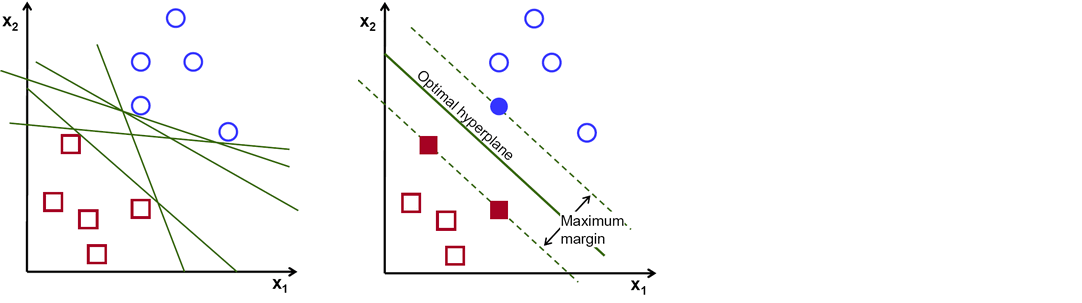
The relationship between the two concepts can be expressed using the formula below:

 Where:

* **ρ(X,Y)** – the correlation between the variables X and Y
* **Cov(X,Y)**– the covariance between the variables X and Y
* **σX**– the standard deviation of the X-variable
* **σY**– the standard deviation of the Y-variable

<https://corporatefinanceinstitute.com/resources/knowledge/finance/covariance/>

# Support Vector Machine and Principal Component Analysis Tutorial for Beginners



Chart, scatter chart

Description automatically generated

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

## Cost Function and Gradient Updates

Text

Description automatically generated

Text, letter

Description automatically generated

Diagram, schematic

Description automatically generated

Text

Description automatically generated

A picture containing text

Description automatically generated

Text

Description automatically generated with low confidence

“Kernel” is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces.

**Gaussian Kernel:**It is used to perform transformation when there is no prior knowledge about data

* **Gaussian Kernel Radial Basis Function (RBF):**Same as above kernel function, adding radial basis method to improve the transformation.

Different SVM algorithms use different types of kernel functions. These functions can be different types. For example**linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.**

Kernel – Given a feature mapping φ, we define the kernel K to be defined as: K(x,z) = φ(x) T φ(z)

**Chart

Description automatically generated**

**Graphical user interface

Description automatically generated with medium confidence**

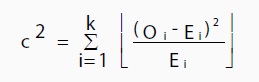
# **Chi-Square Statistic: How to Calculate It / Distribution**

There are **two types of chi-square tests**. Both use the chi-square statistic and distribution for different purposes:

* A**chi-square goodness of fit test** determines if [sample](https://www.statisticshowto.com/sample/) data matches a [population](https://www.statisticshowto.com/what-is-a-population/). For more details on this type, see: [Goodness of Fit Test](https://www.statisticshowto.com/goodness-of-fit-test/).
* A **chi-square test for independence** compares two [variables](https://www.statisticshowto.com/probability-and-statistics/types-of-variables/) in a [contingency table](https://www.statisticshowto.com/what-is-a-contingency-table/) to see if they are related. In a more general sense, it tests to see whether distributions of [categorical variables](https://www.statisticshowto.com/what-is-a-categorical-variable/) differ from each another.

[Back to Top](https://www.statisticshowto.com/probability-and-statistics/chi-square/#top)

A chi-square statistic is one way to show a relationship between two [categorical variables](https://www.statisticshowto.com/what-is-a-categorical-variable/). In statistics, there are two types of variables: [numerical (countable) variables](https://www.statisticshowto.com/quantitative-variables-data/) and [non-numerical (categorical) variables](https://www.statisticshowto.com/qualitative-variable/). The chi-squared statistic is a single number that tells you how much difference exists between your observed counts and the counts you would expect if there were no relationship at all in the population.



You could also use a [p-value.](https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/p-value/) First state the [null hypothesis](https://www.statisticshowto.com/probability-and-statistics/null-hypothesis/#state) and the [alternate hypothesis](https://www.statisticshowto.com/what-is-an-alternate-hypothesis/). Then generate a chi-square curve for your results along with a p-value (See: [Calculate a chi-square p-value Excel](https://www.statisticshowto.com/probability-and-statistics/excel-statistics/calculate-chi-square-p-value-excel/)). Small p-values (under 5%) usually indicate that a difference is significant (or “small enough”).

### Uses

The chi-squared distribution has many uses in statistics, including:

* [Confidence interval](https://www.statisticshowto.com/probability-and-statistics/confidence-interval/)estimation for a population [standard deviation](https://www.statisticshowto.com/probability-and-statistics/standard-deviation/) of a normal distribution from a sample standard deviation [1].
* Independence of two criteria of classification of [qualitative variables](https://www.statisticshowto.com/qualitative-variable/).
* Relationships between [categorical variables](https://www.statisticshowto.com/what-is-a-categorical-variable/) ([contingency tables](https://www.statisticshowto.com/what-is-a-contingency-table/)).
* [Sample variance](https://www.statisticshowto.com/probability-and-statistics/descriptive-statistics/sample-variance/) study when the underlying distribution is normal.
* Tests of deviations of differences between expected and observed frequencies (one-way tables).
* The chi-square test (a [goodness of fit](https://www.statisticshowto.com/goodness-of-fit-test/)test).

# Normalized Data Vs Standardized Data

* Normalization is used when the data doesn't have Gaussian distribution whereas Standardization is used on data having Gaussian distribution.
* Normalization scales in a range of [0,1] or [-1,1]. Standardization is not bounded by range.
* Normalization is highly affected by outliers. Standardization is slightly affected by outliers.

Chart, diagram, box and whisker chart

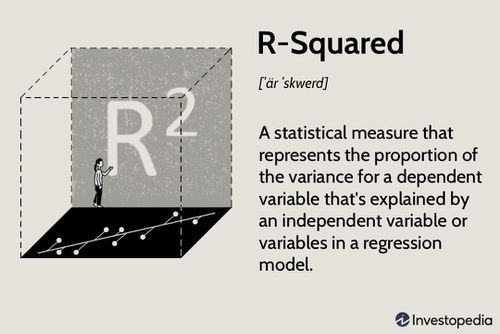
Description automatically generated

Regression

R , Adjusted R , MAE(mean absolute error), MSE (mean square error), RMSE (root mean square error), AIC (Akaike information criterion), BIC (Bayesian information criterion), Residual analysis, Goodness-of-fit test, Cross validation

Classification  Model evaluation methods Accuracy, Confusion matrix, Sensitivity and specificity, ROC (receiver operating characteristic), AUC (area under the curve), Cross validation

General steps to build a model 1. Collecting the data. 2. Preparing the data and fixing issues such as missing values and outliers. 3. Use exploratory analysis to help study the content of your data and select a proper algorithm that suits your need. 4. Training a model using the algorithm you just selected. Start with a simple model that only uses the most important variables/features. 5. Check model performance using the evaluation methods. 6. If the model is not satisfactory, choose another algorithm or introduce different variables into the exsiting model.



R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model. Or:

R-squared = Explained variation / Total variation

# **PCA In Machine Learning - Your Complete Guide To Principal Component Analysis**

<https://www.simplilearn.com/tutorials/machine-learning-tutorial/principal-component-analysis#:~:text=The%20Principal%20Component%20Analysis%20is,plotting%20in%202D%20and%203D>.

The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. It increases interpretability yet, at the same time, it minimizes information loss. It helps to find the most significant features in a dataset and makes the data easy for plotting in 2D and 3D.

The Principal Components are a straight line that captures most of the variance of the data. They have a direction and magnitude. Principal components are orthogonal projections (perpendicular) of data onto lower-dimensional space.

* PCA is used to visualize multidimensional data.
* It is used to reduce the number of dimensions in healthcare data.
* PCA can help resize an image.
* It can be used in finance to analyze stock data and forecast returns.
* PCA helps to find patterns in the high-dimensional datasets.

### Normalize the data

### Standardize the data before performing PCA. This will ensure that each feature has a mean = 0 and variance = 1.s

### Build the covariance matrix Construct a square matrix to express the correlation between two or more features in a multidimensional dataset

### 

### Find the Eigenvectors and Eigenvalues Calculate the eigenvectors/unit vectors and eigenvalues. Eigenvalues are scalars by which we multiply the eigenvector of the covariance matrix.

### 

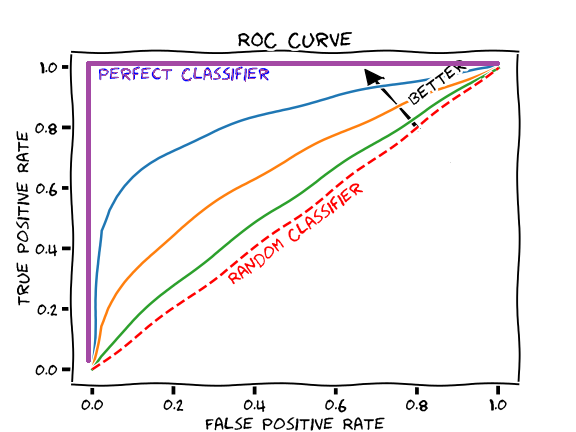
### 4. Sort the eigenvectors in highest to lowest order and select the number of principal components.

Chart, scatter chart

Description automatically generated

Table

Description automatically generated



Steps for calculating test set AUROC for a binary classification task:

1. Train your machine learning model
2. Use the trained model to make predictions on your test set, so that each example in your test set has a classification probability between 0 and 1.
3. Using the model’s output predicted probabilities for the test set, calculate the TPR and FPR for different decision thresholds, and plot a ROC curve.
4. Calculate the area under the ROC curve.

The AUROC is more informative than accuracy for imbalanced data. It is a very commonly-reported performance metric, and it is easy to calculate using various software packages, so it is often a good idea to calculate AUROC for models that perform binary classification tasks.

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated