

Libraries important- Numpy, Pandas, Seaborn, Matplotlib

**Feature Engineering:**

1. Handling missing values:

* Many machine learning algorithms fail if the dataset contains missing values. However, algorithms like K-nearest and Naive Bayes support data with missing values.
* Can lead to lack of precision in the statistical analysis.
* Finding missing values- train\_df.isnull().sum().sum() -> total number in entire dataset
* Ways to handle it: deleting it or replacing them.
* Delete the row with missing values- df.dropna(axis=0)
* Delete the column with missing values- df.drop(['Dependents'],axis=1)
* Filling missing values- df['Dependents'].fillna(0).
* Filling missing value with mean- if outliers are present , it should be handled first- train\_df['LoanAmount'].fillna(train\_df['LoanAmount'].mean())
* Filling with mode value- done with categorical data train\_df['Gender'].fillna(train\_df['Gender'].mode()[0])
* Filling with median value- done in case of outliers- train\_df['Loan\_Amount\_Term'].fillna(train\_df['Loan\_Amount\_Term'].median())
* Replacing with previous value- test.fillna(method=‘ffill')
* Replacing with next value- test.fillna(method=‘bfill')
* Replacing categorical data with most common data by using sklearn-

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy='most\_frequent')

imputer.fit\_transform(X)

1. **Data exploratory methods:**

* Sns.distplot(train[‘age’)
* Sns.countplot(x=’siblings’,data=)- shows the different count of a certain column/category
* Train[‘age’].hist(bins=30,color=’red’)
* Sns.bloxplot(x= ,y= ,data=train)
* Sns.heatmap(train.isnull(), yticklabels=False)

1. Convert categorical data to numerical data-

* Using label encoder-

from sklearn import preprocessing

label\_encoder = preprocessing.LabelEncoder()

df[‘’body\_style”] = label\_encoder.fir\_transform(data[‘body\_style’)

* Convert categorical to numerical value- df[‘sex’].get\_dummies
* Using onehot encoder- if there are k values for categorical data then create k separate features having values 0 or 1.

From sklearn.preprocessing, import OneHotEncoder.

Onehotencoder = OneHotEncoder()

X = onehotencoder.fit\_transform(df)

* Find and replace method
* Label encoding-

obj\_df["body\_style\_cat"] = obj\_df["body\_style"].cat.codes

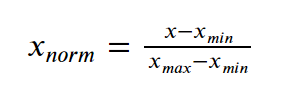
##assigns the code value for each category

1. Detecting outliers:

* Detecting outliers with z-scores: zscore=  (Xi-mean)/std.
* using the Inter Quantile Range(IQR)
* visually summarize the distribution of a variable is the box plot

1. Normalisation and standardization:

* Normalisation formula:



* from sklearn import preprocessing

preprocessing.normalize([x\_array])

* ##normalising the value between 0-2

from sklearn import preprocessing

scaler = preprocessing.MinMaxScaler(feature\_range=(0, 2))

* Standardisation:

from sklearn.preprocessing import StandardScaler

scale.fit\_transform(X\_data)

Feature selection:

1. Pearson’s correlation: summarize the strength of the linear relationship between two data samples. It is the normalization of the covariance between the two variables to give an interpretable score. Code:

from scipy.stats import pearsonr

pearsonr(data1, data2)

1. Spearman’s coefficient: scipy.stats.spearmanr(x, y)
2. In pandas- x.corr(y, method='spearman')

**Points to remember:**

* To find sum of null values in each col:

col for col in df.columns if in df[col].isnull().sum()>1

* To find numerical variables in dataframe:

[col for col in df.columns if df[col].dtypes!=0]

* To find categorical features in dataframe:

[col for col in df.columns if df[col].dtypes==0]

* Grouby use: df.groupby(feature)[‘col values needed’].sum()

df.groupby(year)[‘no. of people dies’].sum()

* Decision trees and tree ensembles can be used with structured data when features are given. Can be used for categorical or numerical data, for classification or regression both.
* Neural networks can be implemented on structured, un- structured or mixed.
* In random forests, random sampling with replacement is done and random set of features are taken to build each tree so that each tree is not identical to each other.

**Machine learning notes:**

There are six types of kernels in SVM:

* Linear kernel - used when data is linearly separable.
* Polynomial kernel - When you have discrete data that has no natural notion of smoothness.
* Radial basis kernel - Create a decision boundary able to do a much better job of separating two classes than the linear kernel.
* Sigmoid kernel - used as an activation function for neural networks.

**What Are Common Bias Types?**

A phenomenon that occurs when an algorithm produces results that are systemically prejudiced/biased due to erroneous assumptions in the machine learning process.

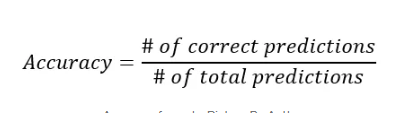
* Anchoring bias,
* Availability bias,
* Confirmation bias,
* Stability bias.

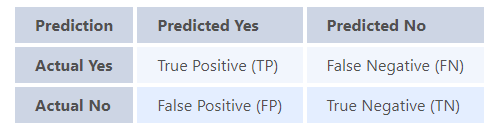
**Difference Between Classification and Regression?**

* Classification is used to classify data into some specific categories. For example, classifying emails into spam and non-spam categories.
* Whereas, regression deals with continuous data and predict the relationship between data   
  For example, predicting stock prices at a certain point in time.

**What is F1 score? How would you use it?**

**Accuracy-** The accuracy metric computes how many times a model made a correct prediction across the entire dataset. This can be a reliable metric only if the dataset is class-balanced; that is, each class of the dataset has the same number of samples.





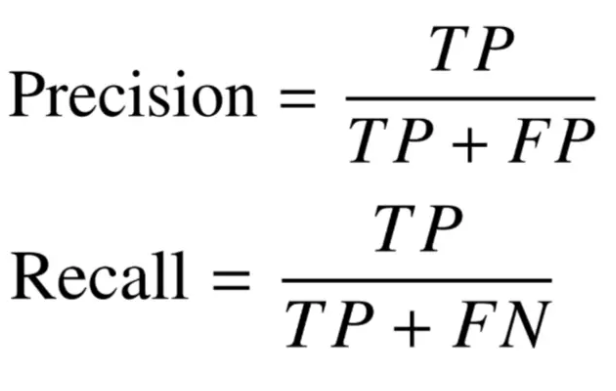
In binary classification we consider the F1 score to be a measure of the model’s accuracy. The F1 score is a weighted average of precision and recall scores.

**Precision**: A precise system is which may not detect all positives but whatever positives are detected are true positives. It does not classify negatives as positives.

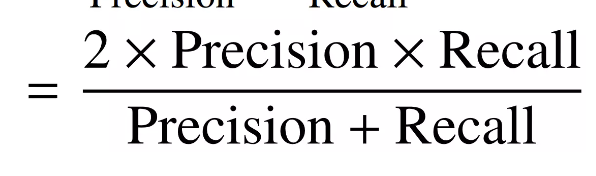
**Recall**:

* A model with **high recall** succeeds well in finding all the**positive cases** in the data, even though they may also wrongly identify some negative cases as positive cases.
* A model with **low recall** is not able to find all (or a large part) of the positive cases in the data.
* Recall in machine learning should be used when trying to answer the question “What percentage of positive classifications was identified correctly?”

Precision recall inversely propotional.



The F1 score is calculated as the harmonic mean of the precision and recall scores. A higher F1 score denotes a better quality classifier.



* A model will obtain a **high F1 score** if both Precision and Recall are high and **low F1 score** if any one of them is low. It is the harmonic mean of recall and precision and tells if any algo is having good precision and recall or not.
* A model will obtain a **medium F1 score** if one of Precision and Recall is low and the other is high

Why is the F1 score calculated using the harmonic mean instead of simple arithmetic or geometric means?

* To put it simply: the harmonic mean encourages similar values for precision and recall. That is, the more the precision and recall scores deviate from each other, the worse the harmonic mean.

**Roc and AUC curve**

**Auc:**

* The AUC quantifies the overall performance of a binary classification model by computing the area under the ROC curve.
* AUC ranges from 0 to 1, where a higher value indicates better model performance.
* An AUC of 0.5 suggests random guessing while an AUC of 1 indicates perfect classification.
* AUC is a useful metric because it measures the model's ability to rank true positives higher than false positives across all possible thresholds.

[**Pipeline**](https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html#sklearn.pipeline.Pipeline) can be used to chain multiple estimators into one. This is useful as there is often a fixed sequence of steps in processing the data, for example feature selection, normalization and classification. Pipeline help as:

* call [fit](https://scikit-learn.org/stable/glossary.html#term-fit) and [predict](https://scikit-learn.org/stable/glossary.html#term-predict) only once on the data to fit a whole sequence of estimators.
* You can [grid search](https://scikit-learn.org/stable/modules/grid_search.html#grid-search) over parameters of all estimators in the pipeline at once. Eg :

Pipeline(steps=[('reduce\_dim', PCA()), ('clf', SVC(C=10))])

param\_grid = dict(reduce\_dim\_\_n\_components=[2, 5, 10], clf\_\_C=[0.1, 10, 100])

grid\_search = GridSearchCV(pipe, param\_grid=param\_grid)

**How to tackle overfitting of data?**

1. [Hold-out](https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d#c287)- Splitting data into train and test data.
2. Cross Validation - diving data into n section and training n-1 sections and checking on that one section. Each time one section which is to be left out changes.



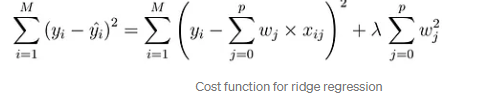
1. [Data augmentation](https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d#f80d)- A larger dataset would reduce overfitting. If we cannot gather more data we can artificially increase the size of our dataset. For example, if we are training for an image classification task, we can perform various image transformations to our image and add to the dataset.
2. Reduce number of [features](https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d#253a)- If we have only a limited amount of training samples, each with a large number of features, we should select the most important features for training.
3. [L1 / L2 regularization](https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d" \l "d178)- Used in regression models. Try increasing lambda so that weights value decrease.

Examples of regularization, includes:

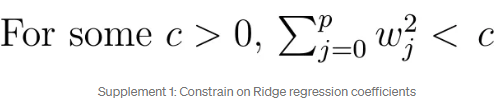
* K-means: Restricting the segments for avoiding redundant groups.
* Neural networks: Confining the complexity (weights) of a model.
* Random Forest: Reducing the depth of tree and branches (new features)

L1 regularization -***Lasso Regression***. Adds “*absolute value of magnitude*” of coefficient as penalty term to the loss function. It shrinks the less important feature’s coefficient to zero thus, removing some feature altogether. this works well for **feature selection** in case we have a huge number of features.

L2 regularization -***Ridge Regression***. Adds “*squared magnitude*” of coefficient as penalty term to the loss function. Loss function for L1 :



i.e. coefficients are restricted. And for large coeff. cost fn. increases. Hence coeff. should be small.



Sum of squared residual (loss function) + lamda\*(square mag of coefficients)

Eg: in linear regression loss fn is : sum of (Yi-XjBj)2 . Now for ridge regresion line loss fn will have added term of lambda. This will minimize the error also. Refer to this link: https://builtin.com/data-science/l2-regularization

[6. Remove layers / number of units per layer](https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d#87f3)  
[7. Dropout](https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d#6f6a)  
[8. Early stopping](https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d#98ac)

**Bias and Variance**

**Bias**

* Bias comes when the model is underfit.
* It is the error rate of the algorithm on the training accuracy.
* Suppose training accuracy is 85% and error rate is 15%. If this 15% can be reduced further, it is called avoidable bias.

**Variance**

* Variance comes when the model overfits the training data.
* It is how much worse the algorithm performs on the test data as compared to training accuracy.
* Suppose the test accuracy is 84% and error rate is 16%, the extra 1% which is coming in test accuracy, can be due to variance, i.e overfitting of the mode.

**Ensemble learning** is a method that combines multiple machine learning models to create more powerful models. The model should always have a balance between bias and variance. By leveraging the diverse strengths of different models, ensemble learning aims to mitigate errors, enhance performance, and increase the overall robustness of predictions. Two of them are:

* Bagging,
* Boosting,

An Outlier is an observation in the dataset that is far away from other observations in the dataset. Tools used to discover outliers are:

* Box plot
* Z-score
* Scatter plot, etc.

**Covariance and Correlation:**

* **Covariance** indicates the direction of the linear relationship between variables.
* **Correlation** measures both the strength and direction of the linear relationship between two variables.
* A correlation between variables, however, does not automatically mean that the change in one variable is the cause of the change in the values of the other variable.
* The value of covariance lies between -∞ and +∞ whereas correlation is betw -1 to 1.

**Boosting vs Bagging**

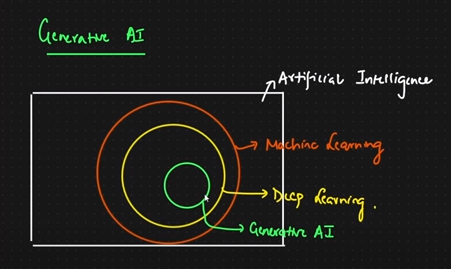
| S.NO | Bagging | Boosting |
| --- | --- | --- |
| 3. | Each model receives equal weight. | Models are weighted according to their performance. |
| 4. | Each model receives an equal weight. | Models are weighed based on their performance. |
| 5. | Training data subsets are drawn randomly with a replacement for the training dataset. | Every new subset contains the elements that were misclassified by previous models. |
| 6. | Bagging applied where the classifier is unstable and has a high variance. | Applied where the classifier is stable and simple and has high bias. |
| 8. | In this base classifiers are trained parallelly. | In this base classifiers are trained sequentially. |
| 9 | Example: The Random forest model uses Bagging. | Example: The AdaBoost uses Boosting techniques |

1. **Bagging** - homogeneous weak learners’ model that learns from each other independently in parallel and combines them for determining the model average.
2. **Boosting** - homogeneous weak learners’ model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.  Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models is added.

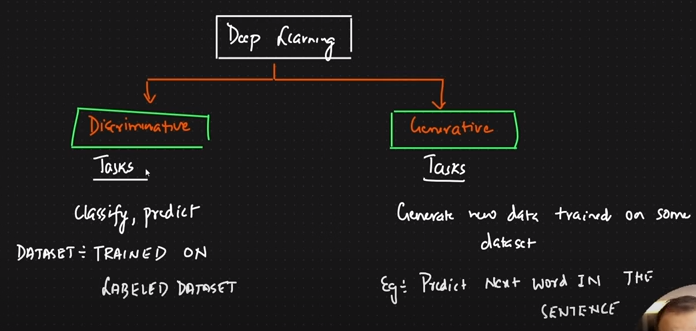
**Decision tree**

* A decision tree is a supervised learning algorithm used for classification and predictive modelling tasks.
* Entropy is calculated at every node to check the quality of split. 0<=entropy<=1. Worst split has entropy =1 when both classes has same number of data. Eg: will it rain- yes(50%) no(50%). Best split has 0 entropy.
* Entropy= -(Po\*log(Po)+ P1\*log(P1))
* Decision tree algorithms are popular in machine learning because they can handle complex datasets with ease and simplicity.
* By asking a sequence of questions and following the corresponding branches, decision trees enable us to classify or predict outcomes based on the data's characteristics.
* Gini impurity: 1- sum ((Pi)^2) ,where Pi=P0, P1.
* 0<=Gini Imp<=0.5.
* Gini impurity is better since it is computationally faster, more efficient since value only goes till 0.5 max.

Generative AI



* It is a subset of deep learning
* Chatgpt, bard- these are LLM(large language models) models, trained on huge huge datasets, based on generative AI.
* Chatgpt is giving output for new data, given trained on some dataset.
* When output is probability, text, number, categories- it is not Gen AI.
* When output is text, image , audio, viedo frmaes- Gen AI.
* Chatgpt 3.5 trained with 170 billion parameters. Trained on internet data



**P- value in statistics:**

**P value gives the probability that the null hypothesis is true in a scenario. We initially take null hypothesis to be true and try to find evidence to go in favour or against this claim. If p value is very less(<0.05), we reject null hypothesis.**

* In statistics, the p-value is the probability of obtaining results at least as extreme as the observed results of a statistical [hypothesis test](https://www.investopedia.com/terms/h/hypothesistesting.asp), assuming that the [null hypothesis](https://www.investopedia.com/terms/n/null_hypothesis.asp) is correct. A smaller p-value means that there is stronger evidence in favor of the alternative hypothesis.
* A p-value of 0.05 or lower is generally considered statistically significant.
* If the p-value is small (0.01 i.e. 1%), it suggests that the observed data has less chances to have occurred by random chance alone under the null hypothesis, which may lead to the rejection of the null hypothesis and favour of an alternative hypothesis.
* The p-value is used to measure the significance of observational data. When researchers identify an apparent relationship between two variables, there is always a possibility that this correlation might be a coincidence. A p-value calculation helps determine if the observed relationship could arise as a result of chance.
* If you compare a 0.04 p-value to a 0.001 p-value. Both are statistically significant, but the 0.001 example provides an even stronger case against the null hypothesis than the 0.04.

**Chi square test**

**When two categorical features are taken into consideration**

* A chi-square test is a statistical test used to compare observed results with expected results. The purpose of this test is to determine if a difference between observed data and expected data is due to chance, or if it is due to a relationship between the variables you are studying. Therefore, a chi-square test is an excellent choice to help us better understand and interpret the relationship between our two categorical variables.

When one continuous variable and one categorical feature with only two categories in it.- T Test

When one continuous variable and one categorical feature but with more than two categories in it.- Anova Test

Two tailed test

* In statistics, a two-tailed test is a method in which the critical area of a distribution is two-sided and tests whether a sample is greater or less than a range of values.
* A hypothesis test that is designed to show whether the mean of a sample is significantly greater than and significantly less than the mean of a population is referred to as a two-tailed test. The two-tailed test gets its name from testing the area under both tails of a [normal distribution](https://www.investopedia.com/terms/n/normaldistribution.asp), although the test can be used in other non-normal distributions.

**AIC**

* Akaike information criterion ( AIC) is a single number score that can be used to determine which of multiple models is most likely to be the best model for a given data set.
* It estimates models relatively, meaning that AIC scores are only useful in comparison with other AIC scores for the same data set.
* The lower AIC score is the better model it is.

**R-squared**

* R-squared also known as the *coefficient of determination*
* **R-squared** is a statistical measure that represents the goodness of fit of a regression model. The value of R-square lies between 0 to 1.
* Where we get R-square equals 1 when the model perfectly fits the data and there is no difference between the predicted value and actual value.
* However, we get R-square equals 0 when the model does not predict any variability in the model and it does not learn any relationship between the dependent and independent variables.
* The value of R-square can also be negative when the model fitted is worse than the average fitted model. (Average fitted model is the y=0 line).
* **Lift Curve:** The lift curve is used to assess the effectiveness of a classification model in terms of its ability to identify the positive class (e.g., rare events or anomalies). It measures the performance improvement of the model over a random baseline.
* **Matthews Correlation Coefficient (MCC):** MCC is a metric that takes into account true positives, true negatives, false positives, and false negatives. It provides a balanced measure of classification performance, particularly in situations where the classes are imbalanced.
* **Logarithmic Loss (Log Loss):** Log loss, also known as cross-entropy loss, is a measure of the accuracy of a probabilistic model's predictions. It is commonly used in classification tasks to evaluate how well the predicted probabilities match the actual class labels.
* **Cook’s distance:** used for regression model.

**Z score:**

<https://www.youtube.com/watch?v=4Fta6KQ1QHQ&list=PLZoTAELRMXVPBTrWtJkn3wWQxZkmTXGwe&index=32>