**Handling Nan values in continuous and categorical data**:

1)Mean/median imputation

2)Sample random imputation.

3)Capturing Nan in new feature.

4)End of Distribution imputation

5) Arbitrary imputation

6) Frequent categories imputation

* Frequent categories Imputation
* Adding new to adding Nan
* Suppose if you have more frequent categories, we just replace NAN with a new category

**Handling Categorical Feature:**

1. One Hot Encoding
2. One Hot Encoding for many categories field.
3. Ordinal Encoding
4. Count Frequency Encoding.
5. Mean Encoding.

**Transformation Technique:**

1. Normalization And Standardization
2. Scaling to Minimum And Maximum values
3. Scaling To Median And Quantiles
4. **Guassian Transformation:**

* Logarithmic Transformation
* Reciprocal Trnasformation
* Square Root Transformation
* Exponential Trnasformation
* Box Cox Transformation

**Handling Imbalanced Data Set:**

1. Under Sampling 🡪imblearn 🡪NearMiss
2. Over Sampler 🡪imblearn 🡪RandomOverSampler
3. SmoteTomek 🡪imblearn 🡪Smotetomek
4. Ensemble Technique🡪imblearn 🡪EasyEnsembleClassifier

**Outliers and Its impact on machine learning:**

1. Which machine learning model sensitive to outliers.

* Naïvye Bayse Classifier. **-🡪 Not sensitive to outlier**
* SVM **🡪 Not Sensitive**
* Linear Regression **🡪Sensitive**
* Logistic Regression **🡪Sensitive**
* Decision Tree Regression and classifier**🡪 Not Sensitive**
* Ensemble (RF, Xgbosst, Gradian Boosting) **🡪 Not Sensitive**
* KNN **🡪 Not Sensitive**
* K means **🡪Sensitive**
* Hierarchal **🡪Sensitive**
* PCA 🡪 **Sensitive**
* Neural Networks 🡪 **Sensitive**

Using IQR and Zscore we can handle Outliers.

**Feature Handling:**

To drop constant feature,

* Variance threshold
* Feature Selection – With Correlation

**Z Score:**

**Z=(xi – U)/std**

**Which is also used to find the probability of remaining area in STD distribution**

**Example: To find probability of student scoring greater than 60 in exam (i.e P(x>60))**

**Probability Density function and CDF- EDA:**

**PDF:**

In this we have how many percentage of data present in y axis and datapoint from data set in X axis. It will be in gaussian distribution.

**CDF:**

In this in y axis it will add up values example if 1st point has 0.1 and 2 point has 2.5 then it add up make it as 3.5 so it won’t be in gaussian distribution the graph will increase.

**Linear Regression:**

Y=mx+c 🡪m = slop and C = intercept

Cost Function = 1/2m(summation from I =0 to M(ycap – y)2

Convergence theorem 🡪 M = m-1(u(sigma)m/dm)\*alpha(it should be smaller number)

**Ridge and lasso:**

**Cost function= Ei🡪m (y-y^)\*2**

**Ridge = Ei🡪m (y-y^)\*2+Y(lambda)\*(slope)\*2**

Ridge is used to reduce the overfitting, in ridge it will shrink the slop never goes to zero

**Lasso = Ei🡪m (y-y^)\*2+Y(lambda)\*|slope|**

Lasso is used to reduce the overfitting as well it will remove feature which is not used, in lasso it will move towards zero and slope may get zero

**Multiple linear regression:**

Y =B0+B1X1+B2X2+B3X3+….

B0=constant

B1X1 🡪 B1 is slope and X1 is feature.

To handle multi collinearity we can remove the feature whose correlation is more than 90, to identify correlation we can use person correlation or OLS from statsmodel.api

**Bias and Variance:**

Low Bias High Variance 🡪 Over fitting

High Bias high Variance 🡪 Under fitting

Low Bias Low Variance 🡪 Balanced model.

**R2 and Adjusted R2:**

R2 = 1 – E(y-y^)2/E(y-yavg)2

Adjusted R2= 1- (1-R2)(N-1)/N-P-1

**Hypothesis Testing:**

1. Hypothesis Test
2. P values
3. Anova Test
4. Z Test

Hypothesis Test, will say the condition is true or not, it will have type 1 error(False Negative) and type 2 error(False Positive)

Process: Initially we will consider null hypothesis as true, then will collect evidence and then done testing and will decide null hypothesis or alternative hypothesis is true. Before testing will consider p value, the P value is 0.05 this is also known as significance value. If value is less then p value will say alternative hypothesis true(reject the null hypothesis) and vice versa.

**T test, Chi square Test, Anova test:**

**When to use which one?**

If the problem statement as only one categorical value we will follow 🡪 One Sample Propositional Test

If the problem statement as two categorical value we will follow 🡪 Chi Square Test

If the problem statement as one continuous or int feature 🡪 T test

If the problem statement as two continuous or int feature 🡪 correlation with T test

If the problem statement as categorical and continuous feature or problem statement with single categorical feature with multiple categories in it 🡪 Anova Test.

**Pvalue and T test:**

T test we have two types,

1. One Sample T test
2. Two Sample T test
3. Paired T test
4. Corelation

**Chi Square Test:**

Alpha = 0.05

Degree of freedom = (no.of.rows – 1)\*(no.of.cols -1)

Chi square formula:

X2 = E(0 – e)2\e

# **Performance Metrics For Classification Problem In Machine Learning**

Confusion Matrix

* FPR (Type 1 Error)
* FNR (Type 2 Error)

Recall (TPR, sensitivity)

Precision (+ve pred val)

To reduce Type 1 error use precision

To reduce Type 2 error use recall

To reduce it in common, need to consider the both

To reduce both type errors we will use Fbeta Score,

Fbeta

F1 = (1+b) precision\*Recall/B2\*Precision+Recall

**Logistical Regression Algorithim:**

First why we no need to use linear alg

To identify the best fit line 🡪 cost function Yi \*WTX

Sigmoid = 1/1+e-x

Logistical Regression algorithm for multi classes 🡪 same like logistic regression working model just include multi class = ‘OVR’

**Decision Tree:**

**Entropy:**

H(S) = - P (+) log2(P+) – P (-) log2(P-)

**Information Gain:**

Gain(S,A) = H(S) – E(From V to Value) |Sv|/|S| \*H(Sv)

Decision tree for numerical variables can be split by using the threshold value it will process for each value in dataset

**Gini Impurity:**

H(S)=1-E(p+2 + p-2)

**KNN (K Nearest Neighbor):**

**To check highly correlated independent features:**

*#with the function we can select highly correlated independent features*

*#it will remove the first feature that is correlated with anything other feature*

def**correlation(df,threshold):**

**coll\_corr**=**set()** *#set of all the names of correlated columns*

**corr\_matrix**=**df**.**corr()**

for**i in range (len(corr\_matrix**.**columns)):**

for**j in range(i):**

if**abs(corr\_matrix**.**iloc[i,j])**> **threshold:**

**colname**=**corr\_matrix**.**columns[i]** *#getting the name of columns*

**coll\_corr**.**add(colname)**

return**coll\_corr**

In [12]:

*#calling the fuction*

**corr\_feature** = **correlation(df**.**iloc[:,:**-**1],0.85)**

**len(set(corr\_feature))**