**Data Wrangling**

---------------

Data Wrangling === Data Transformations

Data Transformation : Converting data from data from 1 to another

: converting the text data to numerical data (encoding)

**Encoding**

-----------

Discrete Categorical (Text) : Nominal, Ordinal

Nominal Data : nominal Encoding

Ordinal Data : ordinal encoding

label encoding

What is dummy variable trap?

if a column consists of n categories, instead of converting to n columns, we have create n-1 columns

why?

reason 1: its reduces time

reason 2: collinearity problem (having high correlation between 2 independent variables)

independent variables : students

dependent variable : trainer

Data Wrangling

-Feature Encoding

-Feature Transformations

-Discretization

-Feature Scaling

Data Analytics : Analyze the given past data, make some observations --- > improve the business

: Descriptive Analytics on Structured Data

Predictive Analytics on Structured Data

steps involved in DA project

1. Business Problem Understanding

2. Data understanding

(i) Collect the data from database

(ii) Understanding each column in detail

(iii) Data Exploration

3. Data Preprocessing

(i) Data Cleaning

(ii) Feature Engineering

(iii) Feature Selection

(iv) Data Wrangling

(v) Train - Test split

4. Modelling (Identify the relation or pattern by applying ML algorithms)

5. Evaluation (calculating accuracy)

6. Model Selection (whether my ML Model is good or bad)

1. Identify the best ML algorithm, which gives highest accuracy in less time

Labelled Dataset : if o/p variable/column is available in given dataset

Unlabelled Dataset : if o/p variable/column is not available in given dataset

How many o/p variables will be in a Dataset : 0 (Unlabelled Dataset)

1 (labelled Dataset)

How many i/p variables will be in a Dataset : 1 to inf

How to identify the o/p variable in given dataset :

2 types of ML techniques

1. Supervised ML technique : Any ML Techniques applied on Labelled Data

Predict o/p variable which is Continuous : Regression

Ex: House Price, Salary

Predict o/p variable which is Discrete : Classification

Ex: Cancer:yes/no Repay/No

2. Unsupervised ML Technique : Any ML Techniques applied on Unlabelled Data

Clustering (groping of similar records)

Ex: Covid (Red,orange,green)

Association Rules (which mostly frequent bought together) MBA : Market Basket Analysis

Recommendation Engines

Cross Sectional Data

-------------------

Regression

Classification

Clustering

Association Rules

Recommendation Engines

Time Series

------------

Time Series Analysis & Forecasting

Data

Training Data is used to train the Machine

(Machine understands the train data & identify the pattern)

testing Data

(apply pattern on testing data & do evaluation)

Training Data (70% - 80%)

if u selected less than 70% in training data : Underfitting Problem

if u selected more than 80% in training data : overfitting problem

Testing (20% -30%)

100 records -- > 40 records (splitted)

Input variables : Students while practising ds

output Variables : trainer

correlation between 2 input variables should be low

correlation between 2 input variables is high --- > colinearity problem

Sol: any 1 drop of them

correlation between 1 input variable & 1 output should be high

correlation between 1 input variable & 1 output is low --- > drop input feature

Train-Test Split

Train Data

-- > We apply ML algorithm on train data & identify the relation (equation) between input & output

-- > using relation, we predict on train data

predict: substituting input data (x\_train) in the equation

--> comparison original output to predict values

(y-train) (ypred\_train)

-- > Train Error (regression) & Train Accuracy (classification)

Ex: i will explain (data) & u will understand (Algorithm) the concepts

exam (data) --- > answers (prediction)

comparison of your answers (prediction) to orginal answers

Test Data

-- > using relation, we predict on test data

predict: substituting input data (x-test) in the equation

--> Test Error (regression) & Test Accuracy (classification)

Bias : train error

variance : test error

Low Bias Low Variance ---- > good Model train error == test error (+-5%)

Low Bias High Variance ---- > bad model (overfitting) train error < test error

Solution to rectifying overfitting : Multiple Solutions

High Bias Low Variance ---- > bad model (underfitting) train error > test error

Solution : Add more Data

High Bias High Variance ---- > Worst

Overfitting : train error < test error

Model performs well on train data, Model doesn't perform on test data

Ex: student performing in class weel, in exam/new\_data student not performing good

sol: Proper Preparation

Underfitting : train error > test error

Model doesn't performs well on train data, Model performs well on test data

Solution to rectify overfitting : Multiple Solutions

Solution to rectify underfitting : Add More Data

in python, lr.fit(x\_train, y\_train)

**Simple Linear Regression**

Simple : 1 input variable & 1 0/p variable

Linear : relation btweeen i/p & o/p is linear

Regression : o/p is continous

Simple : 1 input variable & 1 0/p variable

Non-Linear : relation btweeen i/p & o/p is linear

Regression : o/p is continous

H0 : No difference between average model & regression model (coeficent =0)

{status quo}

RMSE/MSE/SSE avg model <= RMSE/MSE/SSE avg model

H1 : difference between average model & regression model (coeficent != 0)

RMSE/MSE/SSE avg model > RMSE/MSE/SSE avg model

Apply F-test or Anova Test --- > p<0.05 -- > we reject H0

%error, you decreased from avg model to reg model

R2 = How much your regression model is better than average model

Range of R2 : (-inf, 1]

R2 can be -ve : sse of reg > sse of avg

: Error of Regression is higher than error of Average

: Regression failed

R2 can be +ve : sse of reg < sse of avg

: Error of Regression Model is less than error of Average Model

: Regression Working

R2 =1 (perfect model with SSE of reg =0)

**R2 in python can be calculated in 2 ways**

model\_name.score(x,y)

model\_name. score(x\_train, y\_train) ---> train R2

model\_name. score(x\_test,y\_test) --- > test R2

from sklearn.metrics import r2\_score

r2\_score(actual,pred)

r2\_score(y\_train, ypred\_train) --- > train R2

r2\_score(y\_test, ypred\_test) ---- > test R2

Train data

---------

Modelling

prediction

Evaluation ----- > Train R2, Train RMSE (decides based on business problem, default : R2)

Cross Validation

Check whether ur training & cross validation score is same or not

if train R2 == CV Score, proceed with test evaluation

Else, no need to proceed with test evaluation

Possibilities to incease accuracy

- option-1 : Apply other algorithms

- option-2 : by appling better Data Preprocessing

- option-3 : Hyperparameter Tuning

While doing Supervised ML Projects (for Regression & Classification)

Take Jupter Notebook -1 (Data Preprocessing)

1. Business Problem Understanding

2. load data

Data Understanding

Data Exploration

3. Data Preprocessing

Feature Selection

Data Cleaning

Feature Engineering

Data Wrangling

#Reason:

code

# Observation

complete everything before to train-test split (dont select x& y & dont do train-test split)

df.to\_excel("cleaned data.xlsx")

Take Jupter Notebook-2 (LR)

- load the cleaned data

select x & y

train test split

Modellling

Evaluation

Model Seletion

Within good models highest test accuracy is good model

When we apply any ML Algorithm other than Linear regression, it should satisfy 3 conditions

condition 1: train Accuracy == CV Score

condition 2: train accuracy == Test accuracy

condition 3: Its should satisfy business problem requirements

if any 1 condition fails, its called as bad model

When we apply Linear regression, it should satisfy 4 conditions

condition 1: train Accuracy == CV Score

condition 2: train accuracy == Test accuracy

condition 3: Its should satisfy business problem requirements

condition 4: It should satisfy assumptions

why we have to check assumptions in Linear Regression?

Reason : Linear Regression is an assumption model (we are asssuming the relation is linear & simple apply LR)

any dataset, by default we assume its a simple data & simple relation

What are assumptions of linear Regression?

L - Linearity of Errors

I - Idependent of Errors

N - Normality of Errors

E - Equal Variance of Errors (Homoscadescity) & unequal variance of errors (heteroscadescity)

If R^2 (R square) is negative regreesion failed

TV+R+NP -- Sales (relation)

Data Understanding

Data preprocessing

Modelling

- Linear Regression

- for 3 i/p variables, it has generated 3 coefficient values

- whether 3 variables are significant or not

for np : p>0.05 (conclude np is not significant -- > np is not fitting with regression)

-instead of dropping directly the np column, I will influence records

-Im assuming, due to some influential records, np is not getting significant

- Check influential records & I drop influential records

In entire ML, No fixed answer is available

first Model : tv,r,np

influencial records

np is insignificant

second model : drop influencial records

: np is insignificant

third model : drop influencial records & drop np

Simple Regression Project :

Identify the relation between i/p & o/p which gives maximum R2 or minimum RMSE

1 i/p variable (continous or discrete)

1 o/p variable (continous)

steps involved in simple regression project

1. Business Problem Understanding

2. Data understanding

-- > Collect & load the data

-- > Data Exploration

3. Data Preprocessing

-- > Data Cleaning

-- > Data Wrangling

-- > Feature Selection

-- > Identify the best Random state number for Train Test Split

4&5. Modelling & Evaluation

1. First apply LR + calculate Train R2, CV, Test R2 + Check Assumptions

calculate CV

- CV should be applied only on train data

- CV=5 & test\_size=0.2

- scoring ="r2" (if Business Problem : Maximum R2)

scoring ="neg\_mean\_squared\_error" (if business problem : low RMSE)

predict the building price : 2cr

with what accuracy : R2

with what error : RMSE

Calculate Test R2

1. Check every variable p<0.05
2. Check assumptions

if (trainR2 == CV) and (train R2 == test R2) and (All 4 assumptions are satisfied) :

print("consider this as a good model")

else:

print("consider as bad model")

2. Now Apply NLR on same dataset + Calculate TrainR2,CV, TestR2 + Check Assumptions

if (trainR2 == CV) and (train R2 == test R2) and (All 4 assumptions are satisified) :

print("consider this as a good model")

else:

print("consider as bad model")

Assumptions

1. Linearity of Errors (check Scatter plot)

2. Independent of Errors (check whether each & every variable has p<0.05)

3. Normality of Errors (check the skewness)

4. Equal variance of Errors

5. Once all models are completed, then finally identify which is best (which has highest test accuracy)

& Save that algorithm & ignore remaining models

if >1 algorithm has same accuracy:

then select algorithm has completed in less time

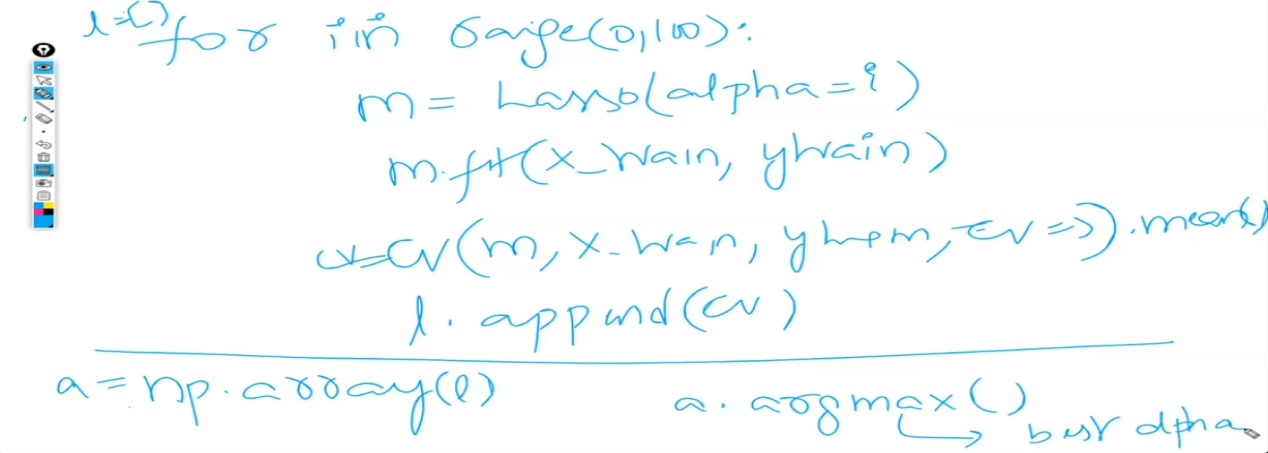
Regularization : by adding penalty term to loss function (add bias) & reducing variance

Loss Function : Sum of Square of errors is loss function ()

Grid Search CV

Grid : Within given data

Search : identify Which gives best value



If we are getting always Alpha value Last number In GridSearchCV then Lasso and Ridge are not working

MULTIPLE LINEAR REGRESSION

Multiple input variables

Business Problem understanding

Data Understanding

Data preprocessing

modelling

for linear regression : assumptions should be satisfied

Evaluation

if R2 +ve : some variables are not significant

when, we have multiple variables which are not significant,

instead of dropping all variables at a time, drop 1 by 1 variable

drop the variable which has highest p-value --- one more time create model & recheck whether other variables are signinficant now.

If other variables are not significant still,then drop now the second variable which has highest p-value

: drop any 1 variable

collinearity Problem : if correlation b/w any 2 input variables is >0.5

check : correlation matrix

Solution

which variable? : Which has highest VIF (variation inflation factor)

Multi collinearity Problem : Multiple variables involved in colinearity problem

check : correlation matrix

Solution

which variable? : Which has highest VIF (variation inflation factor)

: drop any 1 variable

Dummy variable trap?

Ans : instead converting n categories to n columns, we have to convert to n-1 columns

why we have drop 1 category while applying dummy encoding?

Ans: reason 1: it creates colinearity problem

reason 2: when we able to get same information with less columns, no need to create more columns

(reduces processing time)

0/p Variable

------------

Apply only Data Cleaning on o/p variable

wrong data (wrong entry) : replace

wrong data type : convert

Duplicates : drop

missing values : (drop) Never replace Missing values in o/p

Outliers: don't touch them

don't apply Data Wrangling (transformation, scaling, encoding)

don't apply feature selection (transformation, scaling, encoding)

only when you have collinearity problem --- > calculate VIF of each i/p column separately

only when you don't have collinearity problem --- > don't calculate VIF

collinearity: 2 i/p columns having correlation>0.5

Ex: x1, x2 has correlation as 0.6

solution: drop any 1 column of 2 columns which has collinearity problem

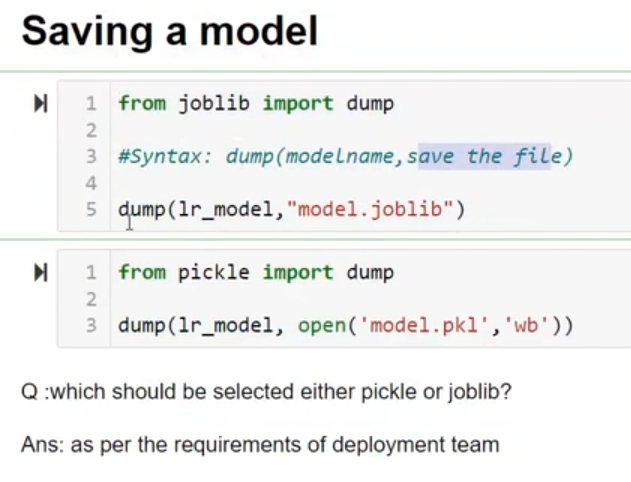
Ex: drop either x1 or x2

which column has to drop? : column which has Highest VIF out of 2

Ex: VIF of x1 = 4.8

VIF of x2 = 9

In this Case, Drop X2



Classification

Its a supervised Machine Learning Technique, where its applied on labelled data for which o/p is discrete

No Project : If o/p variable consists of 1 category

Binary Classification Project : If o/p variable consists of 2 categories

MultiClass Classification Project : If o/p variable consists of >2 categories

Every Classification Algorithms were designed to classify the data

Ex: Stroke Prediction

Evaluation Metrics in Classification Projects

### ****Confusion Matrix****

A table summarizing true positives, true negatives, false positives, and false negatives in classification.

|  | **Predicted Positive** | **Predicted Negative** |
| --- | --- | --- |
| **Actual Positive** | True Positive (TP) | False Negative (FN) |
| **Actual Negative** | False Positive (FP) | True Negative (TN) |

### ****Metrics and Definitions****

**True Positive (TP)**  
Correctly predicted positive cases.

**False Positive (FP)**  
Negative cases incorrectly predicted as positive.

**True Negative (TN)**  
Correctly predicted negative cases.

**False Negative (FN)**  
Positive cases incorrectly predicted as negative.

**True Positive Rate (Sensitivity, Recall, +ve Recall)**  
Proportion of actual positives correctly predicted.  
**Formula**: TPR = TP / (TP + FN)

**False Positive Rate (FPR)**  
Proportion of actual negatives incorrectly predicted as positive.  
**Formula**: FPR = FP / (FP + TN)

**True Negative Rate (TNR or Specificity)**  
Proportion of actual negatives correctly predicted.  
**Formula**: TNR = TN / (TN + FP)

**False Negative Rate (FNR)**  
Proportion of actual positives incorrectly predicted as negative.  
**Formula**: FNR = FN / (TP + FN)

**Precision (+ve)**  
Proportion of predicted positives that are correct.  
**Formula**: Precision = TP / (TP + FP)

**Precision (-ve)**  
Proportion of predicted negatives that are correct.  
**Formula**: Precision = TN / (TN + FN)

**Recall (+ve)**  
Same as True Positive Rate.  
**Formula**: Recall = TP / (TP + FN)

**Recall (-ve)**  
Same as True Negative Rate.  
**Formula**: Recall = TN / (TN + FP)

**Sensitivity**  
Proportion of actual positives correctly identified (same as TPR).  
**Formula**: Sensitivity = TP / (TP + FN)

**Specificity**  
Proportion of actual negatives correctly identified (same as TNR).  
**Formula**: Specificity = TN / (TN + FP)

**ROC Curve**  
 A graph of TPR vs. FPR across different thresholds.

**AU ROC (Area Under ROC Curve)**  
A single value summarizing the performance of the model across all thresholds.

**Misclassification Rate**  
Proportion of incorrect predictions.  
**Formula**: Misclassification Rate = (FP + FN) / (TP + TN + FP + FN)

**Accuracy**  
Proportion of correct predictions.  
**Formula**: Accuracy = (TP + TN) / (TP + TN + FP + FN)

Note : Do not touch output variable,no transformation,no encoding or scaling but we can apply encoding on output in Classification project

Logistics Regression

Q1> To improve accuracy what are options

Q2> How to resolve Over-fitting

Q3> If assumption in Linear regression fails what to do

ANS -> for above 3 questions we have multiple options

1. Better data pre-process
2. Hyper-parameter tuning(not applicable for linear and logistic regression)
3. Regularization
4. Try different algorithm
5. Drop unimportant column(applicable for all)
6. Drop unimportant Record(applicable for linear by checking influential records (unimportant records))

Note - Roc is only applicable for binary class

Note - In Logistic regression higher the AUC better the model

Note -If magnitude are same then no need to apply scaling(very important in KNN because it is dependent on Euclidean Distance therefore if magnitude are not same we apply scaling)

Note - Apply scaling on continuous variable only

Note - Whichever data is from -∞ to +∞ we apply standard scaler

Whichever data is from (0,1) we apply Min max scaler

Classification Projects

Decision Tree

- Based on condition, splitting the data & calssifying the given data

- either based on gini or entropy, we will select the feature for applying condition

- tree type structure & split the data

- tree consists of decision nodes & leaf nodes (1st decision node === root node)

- Advantage:

- easy to understand

- train accuracy is always 100%

- we can identify important features after applying decision tree (every feature importance)

- which feature is not used as a decision node (not used for splitting data), we can drop that feature

- which feature is used as a decision node for multiple times (used for splitting data), more important feature

-important features after applying decision tree (every feature importance will given)

- model. feature\_importances\_ (in python)

- Disadvantage:

- over-fitting (sometimes we get)

- how to overcome this problem?

- optimization (pruning)

Random Forest

- multiple trees with random data

- predict based on voting of multiple decision trees

- each decision tree is not connected to each other (parallel learning)

-each decision tree is completely grown

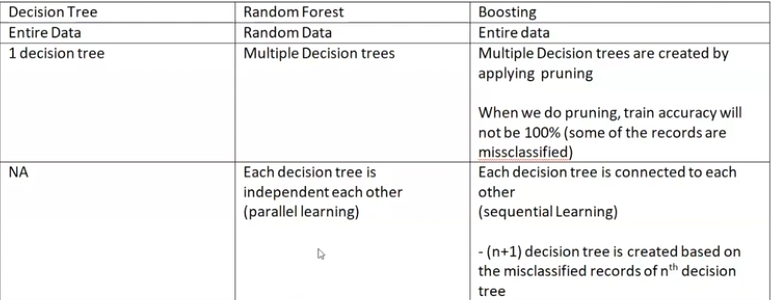
Boosting

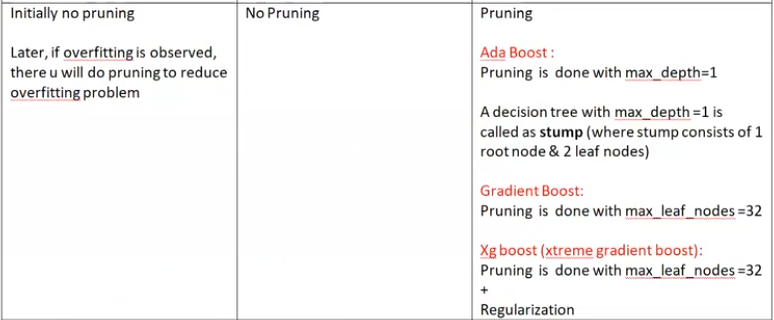
rectifying the errors & getting a better model

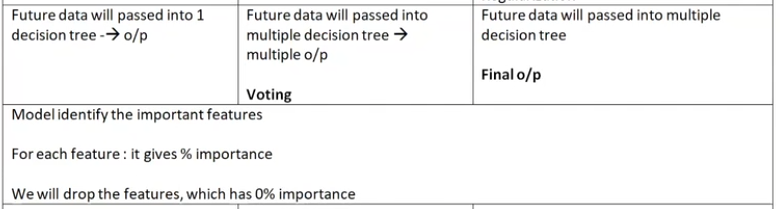
- multiple trees with complete data

- each decision tree is pruned (errors)

- 2nd decision tree will rectify the misclassification







Modelling

> Hyper parameter tuning (best parameter values)

-> Model 1 with best parameter values

> important features

> drop unimportant features

> model2 with best parameters & also important features

for dt, rf, ab, gb, xgb

--------------------

Modelling & evaluation

step-1 : Apply HPT & identify the best parameter & their values

step-2 : using above best\_estimator\_ , calculate feature importances

step-3 : select the features, where importance>0

step-4 : by using the important features along with best parameters, create the final model

Undersampling

reducing the no.of records of majority class

Q : how many records will be considered for majority ?

Ans : no.of records == no.of records of minority class

disadvantage : overall dataset size is reduced

Initial dataset consists of 1,00,000 records

Ex: 99000 records of class 0 (Majority)-- > out of 99000 select randomly 1000 records

1000 records of class 1 (Minority)

after re-sampling, the dataset consists of 2000 records

1000 records of class 0

1000 records of class 1

When to apply undersampling : no.of records of minority class >=500

after apply under-sampling, check total no.of records>100

Minimum sample required to do any analysis : n=30

if n>=30: z-test & t-test (approx)

if n>=1000 : z-test & t-test exactly equal

Dimension Reduction

-----------------  
Dimension : no. Of input features

Reduction : reduce

2types:-

Feature Selection

- Filter Methods (applied in Data Preprocessing stage, applied before modelling)

- Wrapper Methods (linear regression)

- Embedded Methods (after Modelling : features are selected by the model)

Lasso : beta=0 (feature should be dropped)

DT, RF, AB, GB, xGB : model. feature\_importances

: importance =0 (feature should be dropped)

Feature Extraction : PCA (Principal Component Analysis)

: it is applied applied in Data Preprocessing stage, applied before modelling)