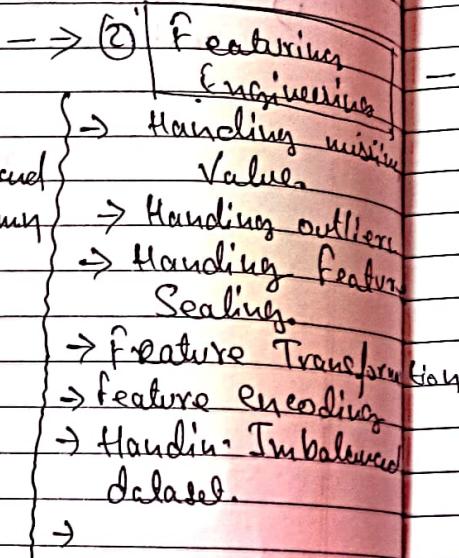
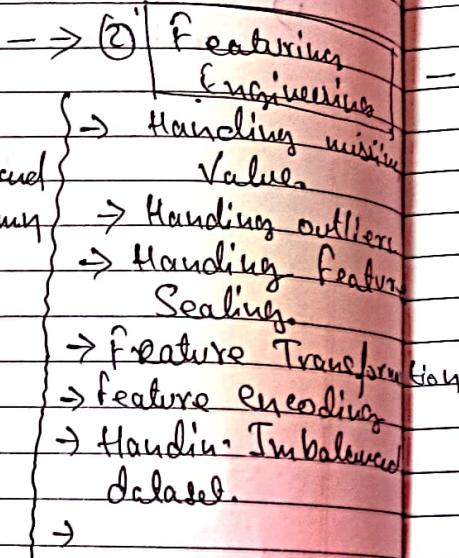


Try with Data Science

① EDA

- Analysis of Data
- Visualization to understand
- Numerical, Categorical column
- Outliers detection
- Finding missing values
- Understand the data



④ Model Training

- All machine learning model.
- Performance metrics.
- Hyper-parameter Tuning
 - Grid Search cv.
 - Random Search cv.
 - Optuna.
 - Tpot
 - Keras Tuner

⑤ Model Deployment

- AWS
- AZURE
- GCP

11/10/22
Tuesday

Date _____
Page _____

Machine learning

Q) What is Machine learning

→ It is the Science of getting Computers to learn and act like humans do and improve their learning over time in autonomous fashion from data and information in the form of observation and real-world interactions.

1. Machine learning Cycle =

a) Data Collection

b) Data Preparation

c) Data Wrangling

d) Data modeling

e) Model Training

f) Model Testing

g) Deployment

a) Data Collection - Goal over here is to gather as much as relevant data as possible.

- Identify various source of inf.

- Gather data

- Combine data from various sources

b) Data Preparation - It deals with exploration of your data to generate far better results.

c) Data Wrangling - It is a process of cleaning & converting raw data into a useable format.

- Filtering / Cleaning up of raw data

- Filtering Noise

- Recognizing & removing outliers

- Removing or filling the missing values

d) Data Modeling - It is the step in which we take the data & select a machine learning algorithm to build model.

- Selecting machine learning algorithm

- Building the models

- Validating the results.

e) Model Training — A machine learning training model is a process in which a ML algorithm is fed with sufficient training data to learn from.

f) Model Testing — This stage of Machine learning life cycle involves checking for the accuracy of the model by providing with the inputs that are unseen.

g) Model deployment — This is the final step in the machine learning life cycle where we have a brilliant model ready to go to production.

2 Artificial Intelligence - It is the science and engineering of making computers capable of performing tasks that typically require human intelligence.

AI classified as -

- Applied A.I (Weak A.I)
- Generalized A.I (Strong A.I)

3) ML is a SubSet of Weak A.I,

Machine Learning - It is a SubSet of A.I which enables machines to learn from past data or experience without being explicitly programmed.

Type of ML -

- i) Supervised learning
- ii) UnSupervised learning
- iii) Reinforcement learning

u Deep Learning :- It is a Subset of M.L
Concerned with the algorithms
inspired by the Structure & function of
human brain (uses of Neural Network)

Ex - Language Translation

colour change of Black & white

12/10/22
Wednesday

Date _____
Page _____

5 Supervised Learning :-

In Supervised learning we train the machine using data which is well "labeled", i.e., some input data is already tagged with correct answer and this algorithm learns from labeled training data that helps us to predict the further outcomes.

The goal of Supervised learning is to map the input Variable (x_1) with the output Variable (y).

6 Types of Supervised ML Algorithms -

i) Regression

- Linear Regression
- Polynomial Regression
- Regression Trees
- Ridge Regression
- Lasso Regression

ii) Classification

- Random Forest
- Decision Trees
- Logistics Regression
- Support Vector Machine
- Naive Bayes

Advantages of Supervised Learning -

- You have full control over what machine is learning
- You can easily test & debug your model
- You can determine the number of classes

1) Disadvantages of Supervised Learning -

- Have Limited Scope.
- Collecting labelled dataset is expensive & time-consuming.
- Wrong Predictions.

NOTE - i) Regression give the Numerical output.

Classification give the Categorical output.
(Ex - Yes/No, Black or White, True/False)

NOTE - • In Supervised Learning, we train the machine using data which is well 'labeled'.

Q When we use Supervised learning?

→ If we have labeled dataset we can use both, but if we have no labeled data we can use only unsupervised.

→ Supervised learning is generally used to classify data or make prediction, whereas unsupervised learning is generally used to understand relationship within data set.

→ Supervised learning is typically done in the context of classification, when we want to map input to output (like and In regression, we want to map input to a continuous output.)

7 Unsupervised Learning - In unsupervised learning, we train the machine using data which is 'unlabelled' and models itself, find the hidden patterns & insights from the given data.

The goals of unsupervised learning is to group unlabelled data according to the similarities, patterns & difference without any prior training of data.

Types of Unsupervised ML Algorithms -

- Clustering
 - Association
- ? this are types

H Unsupervised learning Algorithms -

- i) K-Means Clustering
 - ii) ~~KNN~~ (K-Nearest Neighbor)
 - iii) Hierarchical Clustering
 - iv) Neural Network / Deep learning
 - v) Single Value Decomposition
 - vi) Distribution Models
 - vii) Principal Component Analysis (PCA)
 - viii) Apriori Algorithm
D3SCAN
- ? this are the part of unsupervised learning Algorithms

Advantages of Unsupervised Learning —

- It is used for more complex tasks.
- It's helpful in finding patterns in data.
- Saves lot of manual work & expense.

Disadvantage of Unsupervised Learning —

- Less accuracy.
- Time consuming.
- More the features, More the Complexity.

NOTE - → In unsupervised learning, we train the machine using data which is "unlabeled"

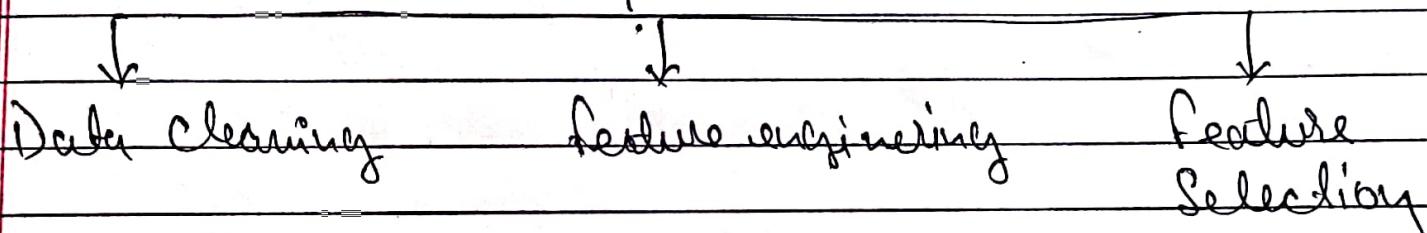
⇒ The biggest drawback in unsupervised learning is that it might result in less accuracy.

(Exploratory Data Analysis) (EDA)

#1 All the lifecycle in a Data Science Project -

- i) Data Analysis / Data - Preprocessing.
- ii) Feature engineering
- iii) Feature Selection
- iv) Model Building
- v) Model Deployment

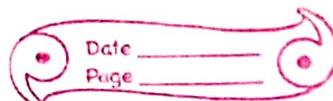
Data-Pre processing



1 Step-wise to do Project -

- i) Understanding about Data Set.
- ii) Import Data Set.
- iii) EDA & Visualization
- iv) Data Cleaning
- v) Feature engineering & Feature Selection
- vi) Model training
- vii) Model Deployment & Predictions.

10/11/22
Saturday



EDA on Advance House Price Prediction

Code ① >>> import libraries

>>> pd.read_csv("path")
(# to see all the features)

Code ② >>> data = pd.read_csv("path")

>>> data.shape

>>> data.head()

Missing values —

missile = write dataset in
place of data only
from this page & after this
page same as it is]

Code ③ # to find missing value of it using graph

>>> data.isnull().sum() [
>>> import missingno as mno
>>> mno.matrix(data)

Code ④ # to get all feature having null value —

>>> feature_with_na = [features for features in data.
.columns if data[features].isnull().sum() != 0]

>>> feature_with_na

Code ⑤ now we will try to get % of null value present
in each feature.

using code no ④ after —

>>> for feature in features_with_na:

>>> print(feature, np.round((data[feature].isnull().
.mean()), 4), "% missing Value")

∴ Note - Don't use features instead of feature if
4 is the decimal value we can change it

Note:- Since there are many missing values, we need to find relationships between missing value & the Sale Price (target feature) with the help of Bar Diagram.

Code :- ⑥ for feature in features with na:

»» data = dataset.copy()

Let make a variable that indicate 1 if the observation or columns was missing (null value) and 0 for no missing value.

[# 1 = missing value

0 = no - missing value.

»» data[feature] = np.where(data[feature].isnull(), 1, 0)

Lets calculate the mean of Sale price & drawing graph where the information is present or missing

»» data.groupby(feature)[['SalePrice']].median().plot.bar()

»» plt.title(feature)

»» plt.show()

To count the feature use -

»» print("Id of houses :- ".format(len(dataset.Id)))

Numerical variable / feature

code ⑥ # list of numerical variable present in the dataset -

»» numerical_feature = [feature for feature in dataset.

• column if dataset[feature].dtype != 'O'

»» len(numerical_feature)

»» dataset[numerical_feature].head()

NOTE:- From the dataset we have 4 feature of year. where we have to extract information from the date time variable like Day of year, year of purchase, year of sold. (Temporal Variable)

code ⑦ »» # list of variables that contain year information -

»» year_feature = [feature for feature in numerical_feature
if 'Yr' in feature or 'Year' in feature]

»» year_feature

code ⑧ # Let explore the content of these Year -

»» for feature in year_feature:

 print(feature, dataset[feature].unique())

Code ② # we will check whether there is a relationship between year of house sold & Sale price.

```
>>> df_hous = df_hous.groupby('YrsSold')[['SalePrice']].median()
>>> df_hous.plot()
>>> plt.xlabel('Year Sold')
>>> plt.ylabel('Median House Price')
>>> plt.title("House Price vs Year Sold")
```

observation - here in the relationship between 'Yrsold' if 'Sale price' is going decreasing which is when the year is increasing which is impossible.

Code ④ # here we will compare the difference between all year features & with Sale price -

```
>>> for feature in year_features:
...     if feature != 'YrsSold':
...         data = df_hous.copy()
...         data[feature] = data['YrsSold'] - data[feature]
...         plt.scatter(data[feature], data['SalePrice'])
...         plt.xlabel(feature)
...         plt.ylabel('Sale Price')
...         plt.show()
```

Now we will try to find the relationships of Sale price with two type of numerical Variable

^{Code no.} 25 # Numeric Variable are of two types -

1) Discrete Variable 2) Continuous Variable -

» discrete feature = [feature for feature in numerical_feature if len(dataSet[feature].unique()) < 25 and feature not in year_feature + ['Id']]

» print ('Discrete Variable count: {}' .format(len(discrete_feature)))

» print (discrete_feature)

OR

» data [discrete_feature].head()

^{Code no.} 26 # Let's find the relationship between discrete-feature & Sale_price with the help of bar plot

» for feature in discrete_feature:

» data = dataSet.copy()

» [data.groupby(feature)[['SalePrice']].median().plot.bar()]

» plt.xlabel (feature)

» plt.ylabel ('Sale feature')

» plt.title (feature)

» plt.show()

Now Same thing for Continuous feature -

Code ②7) Continuous feature = [feature for feature in numerical feature if feature not in discrete feature + year feature + ['J']]

``` print ("Continuous feature : {} ".format (len (continuous feature)))

Code ②8) Now we will draw bar plot same as code no ② but in continuous feature it is difficult to understand the bar plot so we will draw one more graph of histogram (hist).

``` for feature in continuous feature:  
 data = data set . copy ()
 data [feature] . hist (bins=25)
 plt . xlabel (feature)
 plt . ylabel ('count')
 plt . title (feature)
 plt . show ()

Observation → If we will draw the graph we will get to know that most of the features are not 'Normally Distributed'. So, we will try to change or transform the data/graph & it into normally distributed using "Logarithmic Transformation".

Thus from the previous code we get to know that it is not in Normally Distributed. So, we will do Normal-Distribution.

Code 29) # We will use logarithmic transformation for Normally Distribution.

>>> for feature in continuous feature:

>>> data = dataSet. copy ()

>>> if 0 in data [feature]. unique ():

>>> pass

>>> else:

>>> data [feature] = np. log (data [feature])

>>> data ["Sale price"] = np. log (data ["Sale Price"])

>>> plt. scatter (data [feature], data ['Sale Price'])

>>> plt. xlabel (feature)

>>> plt. ylabel ('Sale Price')

>>> plt. title (feature)

>>> plt. show ()

How-to find - Outlier

Note - Mainly Outlier is find through Box-Plot and remember one thing that outlier didn't find in ~~Categorical~~ Categorical feature

Outlier - Box-Plot

Code 50) $\gg>$ for feature in continuous feature :

```

 $\gg> \text{data} = \text{dataset. copy}()$ 
 $\gg> \text{data[feature]} = \text{np. log}(\text{data[feature]})$ 
 $\gg> \text{data. boxplot(feature)}$ 
 $\gg> \text{plt. ylabel(feature)}$ 
 $\gg> \text{plt. title(feature)}$ 
 $\gg> \text{plt. show()}$ 

```

(# this is using normal distribution)

OR

(# if we don't want '0' value in data)

$\gg>$ for feature in continuous features :

```

 $\gg> \text{data} = \text{dataset. copy()}$ 
 $\gg> \text{if } 0 \text{ in data[feature]. unique():}$ 
 $\gg> \quad \text{pass}$ 
 $\gg> \text{else:}$ 
 $\gg> \quad \text{data[feature]} = \text{np. log}(\text{data[feature]})$ 
 $\gg> \text{data. boxplot(feature)}$ 
 $\gg> \text{plt. ylabel(feature)}$ 
 $\gg> \text{plt. title(feature)}$ 
 $\gg> \text{plt. show()}$ 

```

Categorical feature :-

Code (31) $\gg>$ Categorical feature - [feature for feature in dataset.columns if dataset[feature].dtype = 'O']

Code (32) $\gg>$ for feature in categorical feature:
[print ("The categorical of {} feature are : {}").format(feature, len(data[feature].unique()))]

Code (33) # understanding the relationship between the categorical feature & target variable :-

$\gg>$ for feature in categorical feature:
 $\gg>$ dfy = dataset.copy()
 $\gg>$ [dfy.groupby(feature)[['SalePrice']].median().plot.bar()
 $\gg>$ plt.xlabel(feature)
 $\gg>$ plt.ylabel("Sale Price")
 $\gg>$ plt.title(feature)
 $\gg>$ plt.show()

Feature Engineering

- Feature Engineering on the same data set -

Code no. (34) :: Same code no. ①

Code. (35) :: Same code no. ②

Code no. (36) :: # train-test-split -

```
>>> from sklearn.model_selection import train_test_split  
>>> n_train, n_test, y_train, y_test = train_test_split  
(dataset, dataset['SalePrice'], test_size = 0.1,  
 random_state = 0)
```

Missing Value - (Handle categorical)

```
Code no. (37) :: feature_nan = [feature for feature in dataset.columns  
if dataset[feature].isnull().sum() > 1 and dataset  
[feature].dtype == 'O']
```

>>> for feature in feature_nan:

```
>>>     print(f'{feature} : {round((dataset[feature].isnull().mean(), 4))}% missing value')
```

This to find null Categorical feature.

Code ③⑧ # numerical Variable having null value -

```
>>> numerical[with nan = P.feature for feature in
           dataset[P.feature].isnull().sum()
           > 1 and dataset[P.feature].dtype != 'O']
```

>>> for feature in numerical[with nan = P.feature for feature in

```
>>> [print("{} : {}% missing value . format
           (feature, np.round(dataset[feature].isnull(),
           mean(), 2)))
```

Code ⑨ # Replacing the numerical missing value -

>>> for feature in numerical[with nan = P.feature for feature in

```
>>> [median_value = dataset[P.feature].median()
       (# replace by using median since they are
       # outlier)]
```

>>> # Create a new feature to capture nan value -

```
>>> [dataset[P.feature + 'nan'] = np.where(dataset
           [P.feature].isnull(), 1, 0)]
```

>>> [dataset[P.feature].fillna(median_value)
 inplace = True]

>>> dataset[numerical[with nan = P.feature].isnull().sum()