Try with Data set

1. EDA
   - → Analysis of Data
   - → Visulalization to understand
   - → Numerical, Categorical column
   - → Outliers detection
   - → Finding missing values
   - → Understand the data

2. Featuring Engineering
   - → Handling missing Value.
   - → Handling outliers
   - → Handling feature Scaling.
   - → Feature Transformation
   - → Feature encoding
   - → Handling Imbalanced dataset.
   - →

3. Feature Selection
   - → PCA, TSNE
   - → Chi Square test
   - → Anova Test
   - → Correlation
   - → Backward Elimination Technique.
   - → Feature Importance.
   - → VIF technique.

4. Model Training
   - → All machine learning model.
   - → Performance metrices.
   - → Hyper-parameter Tuning
     - → Grid Search cv.
     - → Random Search cv.
     - → optuna.
     - → Test
     - → 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 human 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 for 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
- Recognising & 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 built model
- Selecting machine learning algorithm
- Building the models
- Validing 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 input 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)

ML is a Subset of Weak A.I.

3 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

4 Deep Learning :— It is a Subset of M.L Concerned with the algorithms inspired by the Structure & function of human ~~beir~~ brains. (uses of Neural Network)

Ex — language Translation
colour change of Black & white

12/10/22
Wednesday

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 predit the further outcomes.

The goal of Supervised learning is to map the input Variable (x) 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

Disadvantages of Supervised learning —

- Have limited Scope.
- Collecting labelled dataset is expensive & time-consuming
- Wrong Prediction.

NOTE- i) Regression give the Numeric output whereas
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 learning.

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

⇒ Supervised learning is typically done in the context of classification, when we want to map input to output labels 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 'unlabeled' 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 } this are types
- Association

Unsupervised learning Algorithms —

i) K-Means clustering
~~ii) KNN (K-Nearest Neighbor)~~
iii) Hierarchical clustering
iv) Neural Networks / Deep learning
v) Single Value Decomposition
vi) Distribution Models
vii) Principal Component Analysis (PCA)
viii) Apriori Algorithm
DBSCAN

} 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 manuals 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 results in less accuracy.

Exploratory Data Analysis (EDA)

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

- Data Cleaning
- feature engineering
- feature Selection

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 & Predicting.

19/11/22 Saturday

Date
Page

EDA on Advance house price prediction

Code ① >>> import libraries

>>> pd.pandas.set_option("display.max_columns", None)
(# to see all the features)

Code no ② >>> data = pd.read_csv(r"path")
>>> data.shape
>>> data.head()

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

Missing values —

Code ③ # to find missing value & # using graph
>>> data.isnull().sum()
[>>> import missingno as msno
>>> msno.matrix(dataset)

Code ④ # to get all feature having null value =
>>> feature_with_na = [features for features in data.columns if data[features].isnull().sum()]
>>> feature_with_na

Code ⑤ ∴ now we will try get % of null value present in each feature.
using ~~code~~ 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 ✓
4 is the decimal value we can change it

Note :- Since there are many missing value, we need to find relationship 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()

We make a variable that indicate 1 if the observation or column 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 & draw a graph where the information is present or missing

>>> data.groupby(feature)['SalePrice'].median().plot.bar()
>>> plt.title(feature)
>>> plt.show()

to count the feature name -

>>> 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.
        .columns 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 datetime variable like No. of year, year of purchase, year of sold. (Temporal Variable)

code ⑦ >> # List of variable 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 (23) # we will check whether there is a realation ship between year of house sold & Sale price.

```
>>> dataset.groupby('YrSold')['SalePrice'].median().
    plot()
>>> plt.xlabel('Year Sold')
>>> plt.ylabel('Median House Price')
>>> plt.title("House Price vs YearSold")
```

Observation - here in the relationship between 'YrSold' & 'SalePrice' is going decreasing ~~which is~~ when the year is increasing which is impossible.

Code (24) # here we will compare the difference between all year feature & with Sale price -

```
>>> for feature in year_feature:
>>>     if feature != 'YrSold':
>>>         data = dataset.copy()
>>>
>>>
>>>         data[feature] = data['YrSold'] - data[feature]
>>>         plt.scatter(data[feature], data['SalePrice'])
>>>         plt.xlabel(feature)
>>>         plt.ylabel('SalePrice')
>>>         plt.show()
```

Date
Page

°° Now we will try to find the relationship of Sale price with two type of numerical variable

Code no (25)

```
# Numeric variable are of two types -
# 1) Discreate Variable    2) Continuous variable.

>>> discreate_feature = [ feature for feature in
    numerical_feature if len(dataset[feature].unique())
    < 25 and feature not in year_feature + ['Id']

>>> print ("Discreate Variable count: {}".format(len(
                                    discreate_feature)))

>>> print (discreate_feature)
                    or
>>> data[discreate_feature].head()
```

Code no (26)

```
# Lets find the relationship between discreate-
  feature & Sale_price with the help of bar plot

>>> for feature in discreate_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 discreate_feature + year_feature + ['Id']]

>>> print("Continuous feature {}".format(len(continu_feat

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

histplot

```
>>> for feature in continuous_feature:
        data = dataset.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".

20/11/22 Sunday

Date
Page

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

Code(29) # we will use logarithimic 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-Plo and remember one thing that outlier didn't find in ~~Catego~~ categorical featur

Outlier - (Box-plot)

Code (30)

```
>>> for feature in continuous_feature:
>>>     data = data_set.copy()
>>>     data[feature] = np.log(data[feature])
>>>     data.boxplot(feature)
>>>     plt.ylabel(feature)
>>>     plt.title(feature)
>>>     plt.show()
```

(# this is using normal distribution)

OR

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

```
>>> for feature in continuous_feature:
>>>     data = data_set.copy()
>>>     if 0 in data[feature].unique():
>>>         pass
>>>     else:
>>>         data[feature] = np.log(data[feature])
>>>         data.boxplot(feature)
>>>         plt.ylabel(feature)
>>>         plt.title(feature)
>>>         plt.show()
```

Date
Page

Categorical feature :—

Code (31)
```
>>> categorical_feature = [feature for feature in dataset.columns
                           if dataset[feature].dtype == 'O']
```

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

Code (33)
understanding the relationship between the categorial feature & target variable :—

```
>>> for feature in categorical_feature:
>>>     data = dataset.copy()
>>>     data.groupby(feature)['SalePrice'].median().
        plot.bar()
>>>     plt.xlabel(feature)
>>>     plt.ylabel("Sale price")
>>>     plt.title(feature
>>>     plt.show()
```

Feature Engineering

— Feature Engineering on the same dataset —

Code no (34) ∴ Same code no (1)

Code. (35) ∴ Same code no (2)

Code no (24) # train_test_split —

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

Missing Value ~~Count Category~~

Code no (25)

```
>>> 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("{} : {}% missing value".format
    (feature, np.round(dataset[feature].isnull()
     mean(),4)))
```

// # this is to find null Categorical feature.

Code (38) # numerical Variable having null value -

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

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

Code (40) # Replacing the numerical missing value -

```
>>> for feature in numerical_with_nan:
>>>     median_value = dataset[feature].median()
      (# replace by using median since there are
       outlier)

>>>  # Create a new feature to capture nan value
>>>  dataset[feature+'nan'] = np.where(dataset
                              [feature].isnull(), 1, 0)
>>>  dataset[feature].fillna(median_value,
                             inplace=True)
>>> dataset[numerical_with_nan].isnull().sum()
```