**ADS\_PHASE4 PROJECT SUBMISSION**

**PROBLEM DISCRIPTION:**

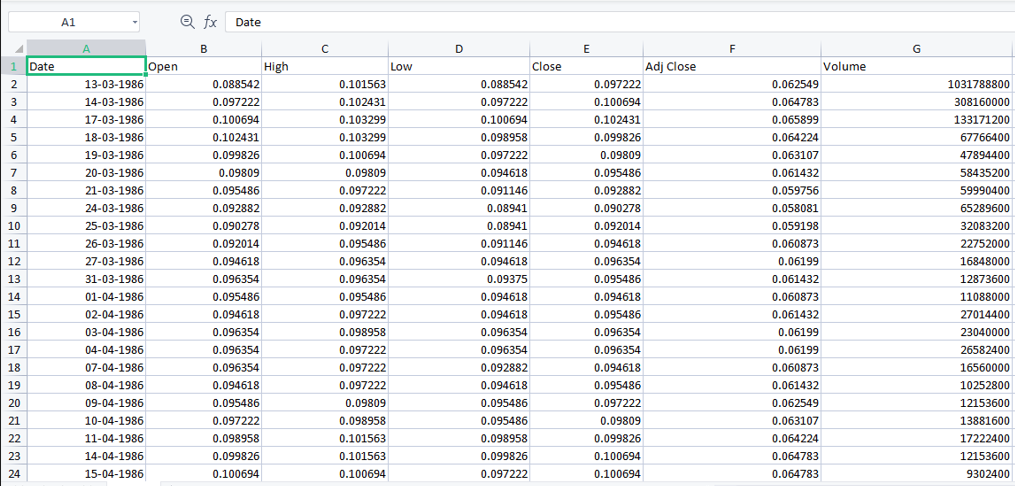
In this part we will continue building your project.

Continue building the stock price prediction model by

* Feature engineering
* Model training
* Evaluation

**Dataset Link:** [**https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset**](https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset%E2%80%AF)

**DATA:**



**Feature engineering**

Feature engineering refers to manipulation — addition, deletion, combination, mutation — of your data set to improve machine learning model training, leading to better performance and greater accuracy.

**Model training**

Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.

**Evaluation**

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring

**DADA BUILDING:**

HyperParameter Optimization

→ Tools for Tuning of datascience and machine learning pipeline

→Configuration Settings

→ Before your training of dataset

→ Is increases your model performance by huge margin

**DIMENSIONALITY REDUCTION**:

Dimensionality reduction is a technique used in data science and machine learning to reduce the number of features (variables or dimensions) in a dataset while preserving or capturing the most important information or patterns. The main goal of dimensionality reduction is to simplify the dataset, improve computational efficiency, reduce noise, and enhance model performance by eliminating irrelevant or redundant features,

1. Better visualization 2. Noise Reduction 3. Overfitting

Effect on Coefficients:

MS

Ridge:

• Ridge regularization tends to shrink the coefficients of less important features towards zero, but it rarely sets

them exactly to zero. As a result, Ridge retains all features in the model.

U,{2665bf71-2bb0-4c6e-943f-8163080f3cde}{142},0.6666666666666666,0.6666666666666666Ridge encourages all features to be of roughly equal importance.

Lasso:

• Lasso regularization has a feature selection property; it tends to set the coefficients of less important features exactly to zero. As a result, Lasso can be used for feature selection.

U,{2665bf71-2bb0-4c6e-943f-8163080f3cde}{179},0.6666666666666666,0.6666666666666666Lasso identifies and selects a subset of the most important features while effectively removing others.

U,{2665bf71-2bb0-4c6e-943f-8163080f3cde}{198},0.6666666666666666,0.6666666666666666Ridge typically produces models with non-zero coefficients for all features. It does not lead to sparse solutions.

U,{2665bf71-2bb0-4c6e-943f-8163080f3cde}{217},0.6666666666666666,0.6666666666666666Ridge is suitable when you believe that most of the features are relevant to the

problem, and you want to reduce multicollinearity.

**•Lasso**:

• Lasso often produces sparse models with only a subset of features having non-zero coefficients.

• Lasso is suitable when you suspect that many features are irrelevant or redundant,

and you want to perform feature selection.

Regularization (Ridge Regression):

1. L2 regularization adds a penalty term to the loss function during training that discourages the model from having large coefficients.

2. The regularization term is proportional to the square of the magnitude of the model's coefficients.

3. It helps to control the model's complexity and can prevent it from becoming too simple.

4. Ridge regression is a common example of 12 regularization.

2.11 Regularization (Lasso Regression):

1. L1 regularization also adds a penalty term to the loss function, but it

discourages the model from having many non-zero coefficients. 2. The regularization term is proportional to the absolute values of the model's coefficients.

3. Lasso regression is a form of L1 regularization that can lead to sparse coefficients, effectively performing feature selection by setting some coefficients to zero.

**L1 Regularization-Lasso Regularization**

Technique to overcome overfitting of data by applying loss function

How??

Adding Penalty Term proportional to the absolute values of the model's Coefficients

What is needed?

> It moves our model to have "Sparse Co-efficient" → by organizing subset of the critical data features and keeping other with ZERO

**PROGRAM:**

import numpy as np

from sklearn. Linear model import Lasso

from sklearn.datasets import make regression

from sklearn.model selection impart train\_test\_split

from sklearn.metrics import mean squared error, r2\_score

Generate some randon data for regression

X, y = make\_regression(samples 100, features, noiset. 1, canoom\_state=42)

MS

E split the data into training and resting sets

X\_train, X\_test, y train, y\_test train\_test\_split( arrays: X,

test\_size=0.2, randon, state=42)

= Create a Lasso regression model with LI regularization alpha = 0.81 W Alpha is the regularization strength, higher values mean stranger regularization lasso\_model = Lasso(alpha=alpha)

# Fit the model to the training data

Lasso\_model.fit(x\_train, y\_train)

= Hake predictions on the test data

Y\_pred = lasso\_model.predict(x\_test)

= Calculate the Mean Squared Error (MSE) to evaluate the model's performance ase = mean

squared error(y\_test,

y\_pred)

U,{39fccc8d-2e8e-4aef-999b-832a2c761b85}{56},0.6666666666666666,0.6666666666666666

DATA

> Cb External Lit

Scratches

RCSV, JSON (Java Script Object Notation), DB (.db SQL, Mongefstructure, unstructure)), image, audio,

video

#File Origin

#1.1 It can be hard used Griles ne will be having in hand)

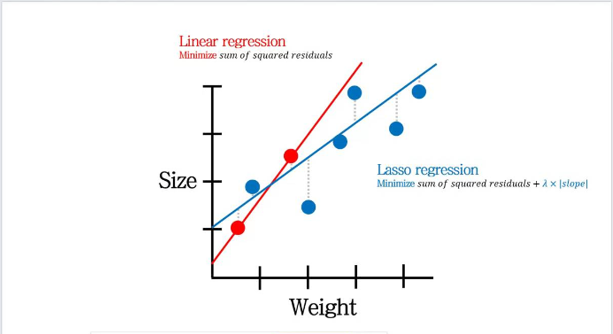
#1.2 It can be in the form of API (URL) -requests (this is the library to nit AP1)

816 No can create a one set of date my ourself for testing

import pandas

my\_data = pandas.read\_csv("Animal Dataset.csv")

print(my\_data)



var

= ["dhoni", "kohti"]

print(var) print(type (var))

print("....................... ")

import pandas

output pandas.Series(var)

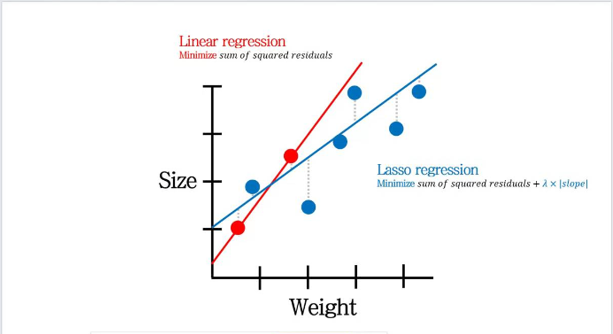
print(output)

print(".......................... ")

output = pandas.DataFrame(var)

print(output1)

print C



#import CKV

#pandas

> Ch External Lit

Scratches

import pandas

bandas.set\_option('display\_max\_o, None)

pandas.set\_option("display\_max\_columns', None) data = pandas.read\_csv("country\_wise\_latest.csv")

print(data)

datal Now deaths 1.fillna (1000, inplace=True)

# print(data)

#print(data.head())

#print(data,tait (10))

#print(data.describe())

print([data.info](http://data.info/)())

**Outliers:**

•Definition: Outliers are data points that significantly deviate from the majority of data points in a dataset. They can be unusually high or low values that do not conform to the expected patterns.

\*Impact: Outliers can distort statistical analyses and machine learning models. They can skew mean and standard deviation, affecting descriptive statistics and making models more sensitive to extreme values.

Detection: Outliers can be detected through statistical methods (e.g., z-scores, IQR), visualization (box plots, scatter plots), or domain knowledge.

\*Handling: Depending on the context, outliers can be removed, transformed, or kept intact. The choice depends on whether the outliers represent genuine data or errors.

THANK YOU