Team 3: Statistics and Linear Regression - Simple & Multiple

Overview: As part of *Statistics and Linear Regression – Simple & Multiple* python package we have implemented statistics, matrix & linear regression modules, below are the APIs description.

S.No	Class	API	Description
1.	Statistics	mean	Calculate mean
2.		variance	Calculate variance & standard deviation
3.		covariance	Calculate covariance & correlation coefficient
4.		correlation_coefficient	Calculate correlation coefficient
5.		correlation_matrix	Calculate correlation coefficient matrix
6.		yule_walker_eq	Solve Yule walker equation
7.		partial_correlation	Calculate partial correlation coefficient
8.		partial_correlation_matrix	Calculate partial correlation coefficient matrix
9.		describe	Statistics of the data – count, mean, variance, standard deviation, min, max
10.	Matrix	add	Add matrices
11.		scalar_multiply	Multiply matrix with a scalar
12.		multiply	Multiply matrices
13.		inverse	Calculate matrix inverse
14.		transpose	Calculate matrix transpose
15.		identity	Generate identity matrix of size n*n
16.		validate	Validate a matrix
17.	LinearRegression	fit	Fit a linear regression model – Simple & Multiple
18.		stats	Get the statistics of the independent & dependent variables
19.		model_stats	Get the statistics of the fitted model
20.		params	Get the parameters of the fitted model
21.		correlation_matrix	Get the correlation coefficient matrix for the input data
22.		partial_correlation_matrix	Get the partial correlation coefficient matrix for the input data
23.		predict	Predict the data, single point
24.		predicts	Predict the data, simultaneously multiple points

Directory & File Structure:

- dataanalytics/stats_linear_regression
  - statistics.py
  - matrix.py
  - linear_regression.py
- tests/stats_linear_regression
  - test_statistics.py
  - test_matrix.py
  - test_linear_regression.py

Data Representation for APIs

Matrix:

1) Example:

1	2	3
4	5	6
7	8	9

mat = [[1,2,3], [4,5,6], [7,8,9]]

2) Example:

1	2
3	4
5	6
7	8

mat = [[1,2], [3,4], [5,6], [7,8]]

3) Example:

1	2	3	4
5	6	7	8

mat = [[1,2, 3,4], [5,6, 7,8]]

Data for Linear Regression:

1) Example: Dependent Variables = 3 , Data Points = 10

X1	X2	X3	Y
1	2	20	50
2	3	22	55
3	2	24	60
4	4	28	65
5	5	30	70
6	4	32	75
7	6	31	80
8	8	33	85
9	6	35	90

Data = [[1,2,3,4,5,6,7,8,9], [2,3,2,4,5,4,6,8,6], [20,22,24,28,30,32,31,33,35]]
y = [50,55,60,65,70,75,80,85,90]

2) Example: Dependent Variables = 1 , Data Points = 4

X1	Y
1	5
2	6
3	5
4	7

Data = [[1, 2, 3, 4]]
Y = [5, 6, 5, 7]

Detailed Documentation

def mean(data: [float]) -> float:			
Property	**Description**		**Example**
Class	Statistics		Calculate mean of 1,2,3,4,5
Method Type	Static Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[float]	Mandatory	[1,2,3,4,5]
Output Parameters	float		3.0
Exception	None & Empty Data		
API Call	mean = Statistics.mean([1,2,3,4,5])		

def variance(data: [float], mean: float = None) -> (float, float):			
Property	**Description**		**Example**
Class	Statistics		Calculate variance and standard deviation of 1,2,3,4,5
Method Type	Static Method		
No of Input Parameters	2		
No of Output Parameters	2		
Input Parameters	[float]	Mandatory	[1,2,3,4,5]
	float	Optional	3.0
Output Parameters	float		2.5
	float		1.5811
Exception	None & Empty Data		
API Call	var, std = Statistics.variance([1,2,3,4,5]) var, std = Statistics.variance([1,2,3,4,5], 3.0)		

def covariance(data_x: [float], data_y: [float], mean_x: float = None, mean_y: float = None) -> (float, float):			
Property	**Description**		**Example**
Class	Statistics		Calculate covariance and correlation coefficient (r) of 1,2,3,4,5 and 5,6,7,8,9
Method Type	Static Method		
No of Input Parameters	4		
No of Output Parameters	2		
Input Parameters	[float]	Mandatory	[1,2,3,4,5]
	[float]	Mandatory	[5,6,7,8,9]
	float	Optional	3.0
	float	Optional	7.0
Output Parameters	float		2.5
	float		1
Exception	None & Empty Data, length of d1 not equal to length of d2		
API Call	cov, r = Statistics.covariance([1,2,3,4,5] , [5,6,7,8,9]) cov, r = Statistics.covariance([1,2,3,4,5] , [5,6,7,8,9], 3.0, 7.0)		

def correlation_coefficient(cov: float, variance_x: float, variance_y: float) -> float:			
Property	**Description**		**Example**
Class	Statistics		Calculate correlation coefficient (r) if covariance = 2.5, data 1 variance = 2.5 data 2 variance = 2.5
Method Type	Static Method		
No of Input Parameters	3		
No of Output Parameters	1		
Input Parameters	float	Mandatory	2.5
	float	Mandatory	2.5
	float	Mandatory	2.5
Output Parameters	float		1
Exception	None Data		
API Call	r = Statistics.correlation_coefficient(2.5, 2.5, 2.5)		

def correlation_matrix(data: [[float]]) -> [[float]]:			
Property	**Description**		**Example**
Class	Statistics		Calculate correlation matrix x1 = [1,2,3,4,5] x2 = [2,4,5,3,2] x3 = [7,6,3,5,8]
Method Type	Static Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[x1,x2,x3]
Output Parameters	[[float]]		[[1.0, -0.121, 0.0822], [-0.122, 1.0, -0.877], [0.0822, -0.877, 1.0]]
Exception	None & Empty Matrix, Invalid Matrix		
API Call	mat = Statistics.correlation_matrix([x1, x2, x3])		

def yule_walker_eq(r12: float, r13: float, r23: float) -> float:			
Property	**Description**		**Example**
Class	Statistics		Calculate partial correlation coefficient (r12,3) if r12 = 0.774, r23= 0.819 r13 = 0.802
Method Type	Static Method		
No of Input Parameters	3		
No of Output Parameters	1		
Input Parameters	float	Mandatory	0.774
	float	Mandatory	0.819
	float	Mandatory	0.802
Output Parameters	float		0.3418
Exception	None data		
API Call	r = Statistics.yule_walker_eq(0.774, 0.819, 0.802)		

def partial_correlation(data_y: [float], data_x:[float], eliminate:[[float]]) -> float			
Property	**Description**		**Example**
Class	Statistics		Calculate partial correlation coefficient (r) if y = [1,2,3,4,5] x1 = [2,5,8,9,6] eliminate x2 =[1,2,4,5,2] x3 = [2,5,6,7,8]
Method Type	Static Method		
No of Input Parameters	3		
No of Output Parameters	1		
Input Parameters	[float]	Mandatory	[1,2,3,4,5]
	[float]	Mandatory	[2,5,8,9,6]
	[[float]]	Mandatory	[[1,2,4,5,2], [2,5,6,7,8]]
Output Parameters	float		-0.799
Exception	None & Empty Data, Invalid eliminate matrix, length of data_y, data_x and data in eliminate matrix if not same		
API Call	pr = Statistics.partial_correlation(y,x1,[x2, x3])		

def partial_correlation_matrix(data: [[float]]) -> [[float]]:			
Property	**Description**		**Example**
Class	Statistics		Calculate partial correlation matrix x1 = [1,2,3,4,5] x2 = [2,4,5,3,2] x3 = [7,6,3,5,8]
Method Type	Static Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[x1,x2,x3]
Output Parameters	[[float]]		[[1.0, -0.103, -0.0507], [-0.103, 1.0, -0.8766], [-0.0503, -0.876, 1.0]]
Exception	None & Empty Matrix, Invalid Matrix		
API Call	mat = Statistics.partial_correlation_matrix([x1,x2,x3])		

def describe(data: [float]) -> {}:			
Property	**Description**		**Example**
Class	Statistics		Describe statistics summary of 1,2,3,4,5
Method Type	Static Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[float]	Mandatory	[1,2,3,4,5]
Output Parameters	{}		{'count': 5, 'mean': 3.0, 'variance': 2.5, 'std': 1.58, 'min': 1, 'max': 5}
Exception	None & empty data		
API Call	mean = Statistics.describe(data)		

def add(m1: [[float]], m2:[[float]]) -> [[float]]:			
Property	**Description**		**Example**
Class	Matrix		Add matrix m1 = [[1,2],[3,4]] m2 = [[4,5],[6,7]]
Method Type	Static Method		
No of Input Parameters	2		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[[1,2],[3,4]]
	[[float]]	Mandatory	[[4,5],[6,5]]
Output Parameters	[[float]]		[[5,7],[9,9]]
Exception	None & empty matrices, Invalid matrix, Matrix dimensions not compatible for addition		
API Call	mat = Matrix.add(m1,m2)		

def scalar_multiply(m1: [[float]], s: float) -> [[float]]:			
Property	**Description**		**Example**
Class	Matrix		Multiply matrix with a scalar m1 = [[1,2],[3,4]] s = 5
Method Type	Static Method		
No of Input Parameters	2		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[[1,2],[3,4]]
	float	Mandatory	5
Output Parameters	[[float]]		[[5,10],[15,20]]
Exception	None & empty matrices, Invalid matrix, None Scalar		
API Call	mat = Matrix.scalar_multiply(m1,s)		

def multiply(m1: [[float]], m2:[[float]]) -> [[float]]			
Property	**Description**		**Example**
Class	Matrix		Multiply matrix m1 = [[1,2],[3,4]] m2 = [[5,6],[0,-2]]
Method Type	Static Method		
No of Input Parameters	2		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[[1,2],[3,4]]
	[[float]]	Mandatory	[[5,6],[0,-2]]
Output Parameters	[[float]]		[5,2],[15,10]]
Exception	None & empty matrices, Invalid matrix, Matrix dimensions not compatible for multiplication		
API Call	mat = Matrix.multiply(m1,m2)		

def inverse(m: [[float]]) -> [[float]]:			
Property	**Description**		**Example**
Class	Matrix		Calculate inverse of matrix m1 = [[1,2],[3,4]]
Method Type	Static Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[[1,2],[3,4]]
Output Parameters	[[float]]		[[-2, 1],[1.5 , -0.5]]
Exception	None & empty matrices, Invalid matrix, Matrix dimensions not compatible for inverse, that is not a square matrix		
API Call	mat = Matrix.inverse(m1)		

def transpose(m:[[float]]) -> [[float]]:			
Property	**Description**		**Example**
Class	Matrix		Calculate transpose of matrix m1 = [[1,2],[3,4]]
Method Type	Static Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[[1,2],[3,4]]
Output Parameters	[[float]]		[[1,3],[2,4]]
Exception	None & empty matrices, Invalid matrix		
API Call	mat = Matrix.transpose(m1)		

def identity(n: int) -> [[float]]:			
Property	**Description**		**Example**
Class	Matrix		Generate identity matrix of size 2*2
Method Type	Static Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	int	Mandatory	2
Output Parameters	[[float]]		[[1,0],[0,1]]
Exception			
API Call	mat = Matrix. identity(2)		

def validate(m:[[float]]) -> [[float]]:			
Property	**Description**		**Example**
Class	Matrix		Validate m1 = [[1,2],[3,4]] is a valid matrix
Method Type	Static Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[[1,2],[3,4]]
Output Parameters	[[float]]		[[1,2],[3,4]]
Exception	None & empty matrices, Invalid matrix		
API Call	mat = Matrix.validate(m1)		

def fit(self, data:[[float]], y:[float]) -> ([{}], [float], [float]):			
Property	**Description**		**Example**
Class	LinearRegression		Fit a linear regression model x1 = [14, 10, 14, 16 10, 10, 14, 16] x2 = [4, 2, 1, 1, 4, 3, 3, 4] x3 = [40, 40, 50, 50, 50, 40, 50, 60] y = [82, 48, 60, 85, 72, 62, 90, 101]
Method Type	Class Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[x1,x2,x3]
	[float]	Mandatory	y
Output Parameters	[{}]		Data Summary statistics [{count, mean, variance, std, min, max, covariance, r, key ,pr (rx1y.x2x3)}]
	[float]		Model Parameters [b1, b2, b3, b0]
	[float]		Predicted values
Exception	None & empty data, Invalid matrix, if number of data points are not same		
API Call	model = LinearRegression() (stats, params, ycap) = model.fit([x1,x2,x3], y)		

def stats(self) -> [{}]:			
Property	**Description**		**Example**
Class	LinearRegression		Get statistics of the independent & dependent variables after fitting a model
Method Type	Class Method		
No of Input Parameters	0		
No of Output Parameters	1		
Input Parameters			
Output Parameters	[{}]		[{count, mean, variance, std, min, max, covariance, r, key ,pr (rx1y.x2x3)}]
Exception	Return None if model is not fitted		
API Call	stats = model.stats()		

def model_stats(self) -> [{}]:		
Property	**Description**	**Example**
Class	LinearRegression	Get fitted model summary statistics – error statistics
Method Type	Class Method	
No of Input Parameters	0	
No of Output Parameters	1	
Input Parameters		
Output Parameters	{}	{count, mean, variance, std, min, max, covariance, r, key}
Exception	Return None if model is not fitted	
API Call	stats = model.model_stats()	

def params(self) -> [float]:		
Property	**Description**	**Example**
Class	LinearRegression	Get the parameters of the fitted model
Method Type	Class Method	
No of Input Parameters	0	
No of Output Parameters	1	
Input Parameters		
Output Parameters	[float]	[b1, b2, b3,…, c]
Exception	Return None if model is not fitted	
API Call	params = model.params()	

def correlation_matrix(self) -> [[float]]:		
Property	**Description**	**Example**
Class	LinearRegression	Get the correlation coefficient matrix for the independent & dependent variables
Method Type	Class Method	
No of Input Parameters	0	
No of Output Parameters	1	
Input Parameters		
Output Parameters	[[float]]	[[yy, yx1, yx2,..], [x1y,x1x1,x1x2,..], .. [x2y,x2x1,x2x2, .]]
Exception	Return None if model is not fitted	
API Call	mat = model.correlation_matrix()	

def partial_correlation_matrix(self) -> [[float]]:			
Property	**Description**		**Example**
Class	LinearRegression		Get the partial correlation coefficient matrix for the independent & dependent variables
Method Type	Class Method		
No of Input Parameters	0		
No of Output Parameters	1		
Input Parameters			
Output Parameters	[[float]]		[[yy, yx1, yx2,..], [x1y,x1x1,x1x2,..], .. [x2y,x2x1,x2x2, .]]
Exception	Return None if model is not fitted		
API Call	mat = model.partial_correlation_matrix()		

def predict(self, point:[float]) -> float:			
Property	**Description**		**Example**
Class	LinearRegression		Predict the dependent variable
Method Type	Class Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[float]	Mandatory	[x1,x2,x3 ..]
Output Parameters	float		y
Exception			
API Call	mat = model.predict([x1,x2,x3 ..])		

def predicts(self, data:[[float]]) -> [float]:			
Property	**Description**		**Example**
Class	LinearRegression		Predict the dependent variable, simultaneously multiple points
Method Type	Class Method		
No of Input Parameters	1		
No of Output Parameters	1		
Input Parameters	[[float]]	Mandatory	[[x1,x2,x3 ..], …. [x1,x2,x3 ..]]
Output Parameters	[float]		[ycap, ycap,ycap ..]
Exception			
API Call	mat = model.predicts([[x1,x2,x3 ..], … [x1,x2,x3 ..]])		