

AI Exercises

Try using **Jupyter notebooks** for doing this.

Numpy

1. Create a numpy array of random float numbers.
2. Convert 1st question result to 'int' datatype.
3. Create a numpy array of random 0 and 1.
4. Convert 3rd question result to a Boolean numpy array.
5. Stack given numpy arrays vertically.
 - a. `a1 = np.array([[1,2],
[3,4],
[5,6]])`
 - b. `a2 = np.array([[7,8],
[9,10],
[10,11]])`
6. Print the indexes that theses tow numpy arrays elements are equal:
`a = np.array([1,2,3,4,5])`
`b = np.array([1,3,2,4,5])`
7. Generate a numpy array of '5' with dimension of 4x7.
8. Generate a numpy array that is 10 repeatation of the giving array.
`a = np.array([[1,2,3], [4,5,6]])`
9. Transpose 8th question result.

Pandas

1. Create a data frame using 'pandas' that have 3 features 'X', 'Y' and 'Z' with random values.
2. Create a dictionary that has 'name', 'age' and 'average_score' features and also 'is_ok' label and then convert it to a pandas data frame.
3. Convert this 2D list to pandas data frame.
`lists = [[2, 'Vishal', 22],
[1, 'Kushal', 25],
[1, 'Aman', 24]]`
4. Load 'diabetes_prediction_dataset.csv' and print the head of this data.

- Sort 4th question loaded data frame by 'age' and 'blood_glucose_level' features then print the result.
- Calculate the mean and mod and min and max of 'bmi' feature value of the 5th question result by aggregate functions of pandas data frames.
-

Confusion Matrix

- Create confusion matrix using given data.
actual = numpy.random.binomial(1, 0.9, size = 1000)
predicted = numpy.random.binomial(1, 0.9, size = 1000)

Regression

- Draw an scatter plot using matplotlib in python and show the Linear Regression of given data.
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

Final Project

In this project you are going to learn and practice topics such as:

- ✓ Numpy
- ✓ Pandas
- ✓ Confusion Matrix
- ✓ Normalization
- ✓ Training
- ✓ Testing
- ✓ Train Test Data
- ✓ Cross Validation

Question:

Try to create a Perceptron model and then train it using '**diabetes_prediction_dataset.csv**' data and finally test it. At the end you must do Cross Validation on this model and then draw the Confusion Matrix using results of predictions.

Som hints:

- ❖ First of all you must normalize and also digitize your data. If you couldn't do that, then use the 'normalized_data.csv' data.
- ❖ For creating perceptron model you can use this code:
 - `from sklearn.linear_model import Perceptron`
 - `perceptron = Perceptron()`
- ❖ For training model you can use this code:
 - `perceptron.fit(X_train, y_train)`
- ❖ For Testing your model you can use this code:
 - `y_pred = perceptron.predict(X_test)`
- ❖ For doing Cross Validation you can use this code:
 - `from sklearn.model_selection import cross_val_score`
 - `cv_scores = cross_val_score(model_name, X_data, y_data, cv=cross_validation_folds)`