

Session 11

Friday, March 31, 2023 6:17 PM

Match Columns

1. Lazy Learner

2. Eager learner

A. Prediction is fast

B. Prediction is slow

1. Lazy learner is an algorithm that postpones the actual processing of the data until it is absolutely necessary. In other words, it doesn't perform any computation until a query is made to it. One example of a lazy learner is the k-nearest neighbors algorithm, which stores all the training data and only looks for the k closest neighbors when it needs to make a prediction.
2. Eager learner is an algorithm that eagerly builds a model using all available training data, and then uses that model to make predictions or classifications. In other words, it processes all the data upfront and creates a model, which can be used to classify or predict future data. One example of an eager learner is the decision tree algorithm, which creates a tree-like model to classify data based on pre-selected features.

Lazy learners are generally faster than eager learners in terms of training time because they defer computation until a query is made. However, they can be slower during prediction time because they need to compute the result on the fly. On the other hand, eager learners are slower during training because they need to process all the data upfront to build the model. But, they are faster during prediction time because they can simply apply the model to new data. So, the choice between a lazy and an eager learner depends on the specific use case and the tradeoffs between training time and prediction time.

Yes, in general, prediction with an eager learner is faster than prediction with a lazy learner because the eager learner has already processed all the data and built a model which can be applied quickly to new data. The lazy learner, on the other hand, needs to do some computation on the fly in order to make predictions, which can be slower. However, this is a generalization and the actual performance of the learners depends on many factors, including the size and complexity of the data and the specific algorithm used.

You have a 10000 * 500 pandas data frame. I give you name of the columns(500 names) like this – name1, name2, name3, name4.....name5. How you will take these names from notepad and make it your column names?

```
columns_name=["name"+str(i) for i in range(1,501)] print(columns_name)
```

Emp Id	Name	Salary	Dept
1	Aman	100	IT
2	SRD	120	HR
3	Gaurav	90	IT
4	Ankit	130	IT
5	Imran	105	HR

Write a query to find top second highest salary from IT department

```
"SELECT * FROM EMP T WHERE T.Dept=="IT" ORDER BY T.Salary DESC LIMIT 1,1"
SELECT * FROM emp WHERE Dept = 'IT' ORDER BY Salary DESC LIMIT 1 OFFSET 1;
```

Emp Id	Name	Salary	Dept
1	Aman	100	IT
2	SRD	120	HR
3	Gaurav	90	IT

Write a query to find all records where first character of name is vowel

3	Gaurav	90	IT
4	Ankit	130	IT
5	Imran	105	HR

select * from emp where left(name,1) in ('a','e','i','o','u')

Find 70th Percentile of these numbers – don't use tool 😊

78, 54, 34, 21, 65, 32, 90, 54, 87, 32, 4, 11

4, 11, 21, 32, 32, 34, 54, 54, 65, 78, 90

$$n = 11$$

$$\text{percentage} = \frac{70}{100} \times 11 = 8.4$$

= 9th term

- Calculate the pth percentile location:

$$i = \frac{P}{100}(n)$$

- Determine the percentile's location and its value.
- If i is a **whole number**, the percentile is the average of the values at the i and (i+1) positions
- If i is **not a whole number**, the percentile is at the (i+1) position in the ordered array

Give practical example of different types of Bias in data

Bias data is data that has systematic errors or inaccuracies that can impact the results of data analysis or machine learning models. Here are some examples of biased data:

1. **Sampling bias:** This occurs when the sample of data is not representative of the population it is meant to represent. For example, if a study on customer satisfaction only collects data from customers who have had positive experiences with a product or service, the results will be biased towards positive reviews.
2. **Measurement bias:** This occurs when the data collection methods or instruments are flawed or inaccurate. For example, if a survey on fitness levels only asks participants to self-report their exercise habits, the results may be biased due to participants overestimating or underestimating their activity levels.
3. **Confirmation bias:** This occurs when the data is chosen or analyzed in a way that confirms pre-existing beliefs or hypotheses, rather than being analyzed objectively. For example, if a study on the effectiveness of a drug only focuses on positive results and ignores negative results, the results may be biased towards the drug being more effective than it actually is.
4. **Social bias:** This occurs when the data reflects societal biases or discrimination. For example, if a hiring algorithm is trained on data that predominantly consists of male applicants, the algorithm may be biased against female applicants.
5. **Temporal bias:** This occurs when the data is collected or analyzed during a specific time period, which may not be representative of the current situation. For example, if a study on the prevalence of a disease only looks at data from the 1980s, the results may not accurately reflect the current situation.

It's important to identify and address bias in data to ensure that data analysis and machine learning models are accurate and unbiased.

Find maximum number of a list without using in built function.

To write a function to return the maximum number from a list without using Python max() function, you can loop through the list and compare each element to the maximum number found so far. Here's an example function:

```
def find_max(numbers): max_num = numbers[0] for num in numbers: if num > max_num:
max_num = num return max_num
```

In this function, we first set the max_num variable to the first element in the list. Then, we loop through the list and compare each element to the max_num variable. If an element is greater than max_num, we update max_num to be that element. Finally, we return the max_num variable, which should contain the maximum number in the list.

You can use this function as follows:

```
numbers = [3, 5, 1, 9, 2, 8] max_num = find_max(numbers) print(max_num) # Output:
9
```

This will print out the maximum number in the numbers list, which is 9.

Mystring = "my name is Aman Kumar"

Delete "m" from "Aman" in above string

```
Mystring = "my name is Aman Kumar"
l = Mystring.split()
l[3]=l[3].replace("m","")
new_name=" ".join(l) print(new_name)
```