Session 10

Monday, April 3, 2023 11:41 PM

Activation function	Output range
1. Regression Models	A. High bias
2. Ensemble Models	B. Low Bias

Ensemble models and regression models differ in their ability to handle bias in data.

Regression models, such as linear regression, can be highly sensitive to bias. This is because regression models attempt to find a single line of best fit through the data, and if the data is biased, this line may not accurately represent the underlying relationship between the variables. As a result, regression models can produce biased predictions, leading to inaccurate results.

Ensemble models, on the other hand, are designed to reduce bias and improve prediction accuracy. Ensemble models combine the predictions of multiple models to create a final prediction, effectively averaging out any bias in the individual models. This approach can help to mitigate the impact of any bias in the data, resulting in more accurate predictions and better model performance.

However, it is worth noting that ensemble models can still be impacted by bias if the individual models used to create the ensemble are also biased. Therefore, it is essential to carefully select and train the individual models to ensure that they are not themselves biased.

In summary, while regression models can be highly sensitive to bias in data, ensemble models are designed to mitigate the impact of bias and improve prediction accuracy. By combining the predictions of multiple models, ensemble models can create more accurate predictions, making them a valuable tool in data analysis and machine learning

Parametric models are a type of statistical model that relies on a specific set of assumptions about the underlying distribution of the data. These models assume that the data follows a particular probability distribution, such as a normal distribution, and use this information to estimate the parameters of the distribution (such as the mean and standard deviation). Linear regression is an example of a parametric model, as it assumes that the relationship between the independent and dependent variables is linear and follows a normal distribution.

Parametric models can be sensitive to bias in the data, as they rely on specific assumptions about the underlying distribution. If the data is not normally distributed or does not follow the assumed distribution, the estimated parameters may be biased, resulting in inaccurate predictions. Additionally, parametric models can be sensitive to outliers, as these can have a significant impact on the estimated parameters of the distribution.

However, parametric models can also have benefits, such as being more interpretable and providing insights into the underlying distribution of the data. They can also be more efficient and require less data than non-parametric models. In some cases, parametric models may even be the preferred choice if the assumptions of the model hold true and the data is well-behaved.

In summary, parametric models can be sensitive to bias in the data due to their reliance on specific assumptions about the underlying distribution. However, they can also have benefits such as interpretability and efficiency, and may be the preferred choice in certain situations where the assumptions hold true.

1. Overfitting	A. High Bias
2. Underfitting	B. High Variance

Overfitting and underfitting are two common problems in machine learning that are closely related to bias and variance. Bias refers to the difference between the predicted values and the true values, while variance refers to the amount that the model's predictions vary for different training datasets.

Overfitting occurs when a model is too complex and captures the noise in the training data, leading to high variance and low bias. This means that the model is able to fit the training data well, but it does not generalize well to new data. The model may have poor performance and accuracy when tested on the validation or test set.

Underfitting occurs when a model is too simple and does not capture the underlying patterns in the data, leading to high bias and low variance. This means that the model is not able to fit the training data well, and may have poor performance and accuracy on both the training and validation sets.

To address overfitting, we can reduce the complexity of the model, for example by using regularization techniques or reducing the number of features. This can help to reduce the variance and improve the model's ability to generalize to new data.

To address underfitting, we can increase the complexity of the model, for example by adding more features or increasing the model's capacity. This can help to reduce the bias and improve the model's ability to fit the training data.

In summary, overfitting and underfitting are related to the bias and variance of the model. Overfitting leads to high variance and low bias, while underfitting leads to high bias and low variance. To address these problems, we can adjust the complexity of the model to balance bias and variance and improve the model's performance.

JOIN and SUBQUERY are two common techniques used in SQL to combine data from multiple tables.

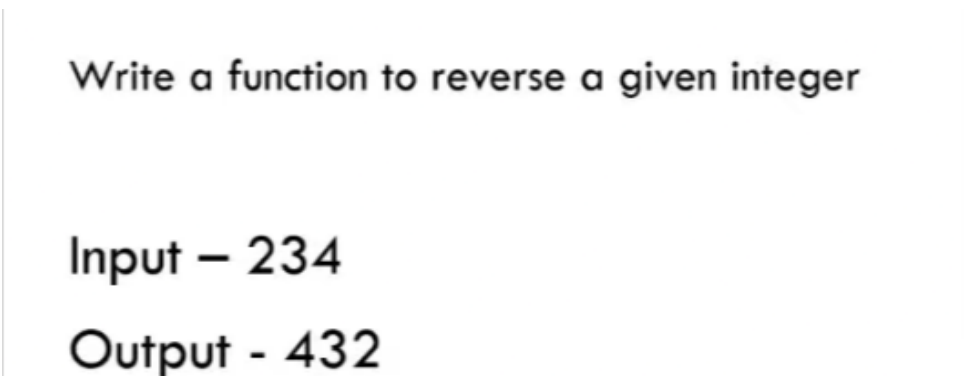
JOIN is used to combine rows from two or more tables based on a related column between them. The most common types of JOIN are INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN. JOIN can be used to combine multiple tables into a single result set and can be used for complex filtering and sorting operations.

In contrast, SUBQUERY is a query nested inside another query. A subquery can be used to retrieve data from one or more tables and use that data as a filter or to perform further calculations in the main query. A subquery can be used to simplify complex queries and perform more advanced filtering and sorting operations.

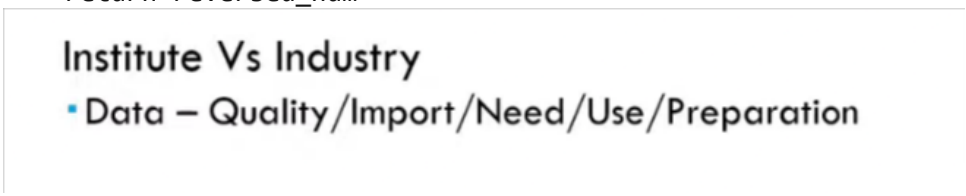
Here are some differences between JOIN and SUBQUERY:

- JOIN is used to combine rows from two or more tables, while SUBQUERY is used to retrieve data from one or more tables and use that data as a filter in the main query.
- JOIN can be used to combine multiple tables into a single result set, while SUBQUERY is used to perform more complex filtering or calculations on a single table or a subset of a table.
- JOIN can be more efficient than SUBQUERY for combining large amounts of data from multiple tables, while SUBQUERY can be more efficient for filtering or calculating data from a single table or a small subset of data.

In summary, JOIN and SUBQUERY are both useful techniques for combining data in SQL, but they are used for different purposes. JOIN is used to combine data from multiple tables, while SUBQUERY is used to perform more complex filtering and calculations on a single table or a subset of data.



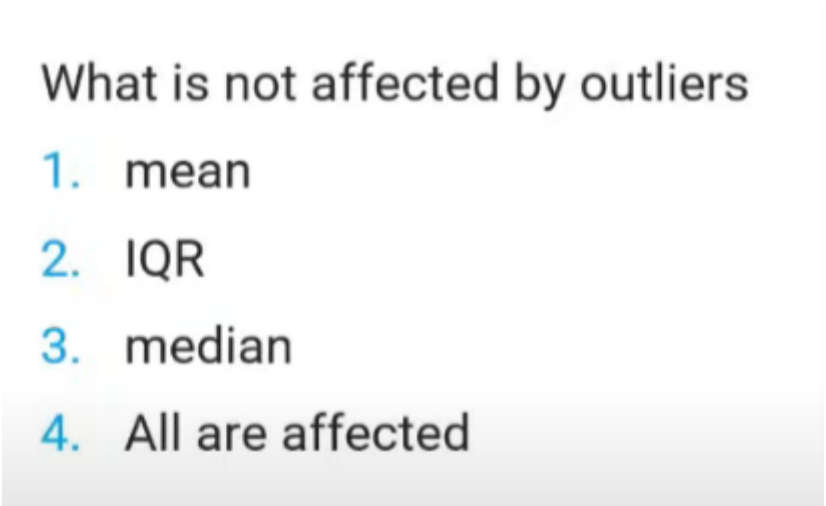
```
def reverse_digit(num):

    # Convert the number to a string and reverse the characters

    reversed_num_str = str(num)[::-1]

    # Convert the reversed string back to an integer
    reversed_
num = int(reversed_num_str)

    # Return the reversed integer
    return reversed_num
```



Institutes and industries are two different environments that offer different experiences and opportunities for data scientists. Here are some key differences between data science tasks in an institute and in an industry setting:

1. **Research vs Production:** In an institute setting, data scientists are typically involved in research-oriented tasks such as developing new algorithms, testing new models or improving existing models. In contrast, in an industry setting, data scientists are more focused on applying existing models and algorithms to solve real-world problems and produce valuable business insights.
2. **Data Size and Complexity:** In an industrial setting, the amount and complexity of data are often much larger and more challenging to deal with than in an institute setting. This can require a greater degree of data engineering and cleaning to prepare the data for modeling, as well as more advanced modeling techniques to analyze the data.
3. **Collaboration:** Institutes often have a more collaborative environment, with data scientists working in teams to develop research projects, while in industry, data scientists may work more independently, collaborating with other teams such as product managers, engineers and business analysts to solve specific business problems.
4. **Business Impact:** In an industry setting, the focus is on producing business value through data insights, while in an institute setting, the focus is on producing new knowledge and contributing to academic research. This means that data scientists in industry must be able to understand the business context of their work and communicate their findings effectively to non-technical stakeholders.

In summary, while both institutes and industries provide opportunities for data scientists, the focus and tasks differ significantly, with research-oriented tasks in institutes and value-driven tasks in industry settings.



The correct answer is **B. median**.

Outliers are values in a dataset that are significantly different from the other values. They can have a significant impact on various statistical measures such as the mean, standard deviation, and range.

However, the median is a robust measure of central tendency that is not affected by outliers. The median is defined as the middle value in a sorted dataset, and it is not influenced by extreme values.

On the other hand, the mean is sensitive to outliers because it takes into account all the values in the dataset. A single outlier can significantly skew the mean towards the outlier value.

The interquartile range (IQR) is a measure of dispersion that is also resistant to outliers. It is defined as the difference between the 75th percentile (Q3) and the 25th percentile (Q1) of a dataset. Since it only considers the middle 50% of the data, it is not influenced by extreme values.

Therefore, the correct answer is B. median.

What are various ways to control overfitting in -

1. Regression models
2. Ensemble models/DT
3. Deep neural network"

There are several ways to control overfitting in different types of models:

1. Regression models:
   Regularization techniques such as L1 or L2 regularization can be used to add a penalty term to the loss function, which helps in reducing the magnitudes of the coefficients and prevent overfitting.
- Cross-validation can be used to evaluate the model and select the appropriate hyperparameters.

2. Ensemble models/DT:
- Bagging techniques such as random forest can be used to reduce the variance of the model by averaging the predictions of multiple decision trees.
- Boosting algorithms such as gradient boosting can be used to iteratively improve the performance of the model by combining weak learners into a strong learner.

3. Deep neural network:
- Dropout regularization can be used to randomly drop out some nodes during training, which prevents the network from relying too heavily on a few input features and improves generalization.
- Early stopping can be used to stop the training process once the validation error starts increasing, preventing the network from overfitting to the training data.
- Batch normalization can be used to normalize the activations of the network, which helps in reducing internal covariate shift and improving gradient flow, thereby preventing overfitting.

Write a code snippet to give index of second odd number from a list

Input – [33,32,31,44]
Output – 2

```
Input = [33, 32, 31, 44]
count_odds = 0
for i in range(len(Input)):
    if Input[i] % 2 != 0:  # check if the number is odd
        count_odds += 1
        if count_odds == 2:  # check if we have found the second odd number
            print(i)  # print the index of the second odd number
            break
```

Delete a list of keys from a dictionary

Mydict = {"name":"Aman", "Roll no":4, "city":"London", "channel":"UFDS"}

Remove keys [“Roll no”, “channel”]

```
mydict = {"name": "Aman", "Roll no": 4, "city": "London", "channel": "UFDS"}
keys_to_remove = ["Roll no", "channel"]

for key in keys_to_remove:
    if key in mydict:
        del mydict[key]
print(mydict)
```