Data Leakage

"***A scenario when ML model already has information of test data in training data, but this information would not be available at the time of prediction, called data leakage. It causes high performance while training set, but perform poorly in deployment or production."***

Why does data Leakage happen?

Now, let's understand the reason for data leakage in a better manner.

Train-Test contamination:

- Initially, the input dataset is split into two different data sets, i.e., training and test data sets (sometimes validation set also), and there are some possibilities that some of the data in the train set is present in the test set and vice versa, i.e., both the test set may share some same information.
- In this case, when we train our model, it gives outstanding results on both data sets with high accuracy but as soon as we deploy it to the production environment, it does not perform well because when a new data set/ completely unseen data is applied, it won't be able to handle it.

Data Preprocessing and Feature Engineering :

Missing Values(Nan)

```
 Start
|
Calculate the mean of the training data for each feature
|
Calculate the mean of the test data for each feature
|
Replace all missing values in the training data with the corresponding train mean
|
Replace all missing values in the test data with the corresponding test mean
|
End
```

Normalization(scaling)

Target leakage

Target leakage occurs when your predictors include data that will not be available at the time you make predictions. It is important to think about target leakage in terms of the *timing or chronological order* that data becomes available, not merely whether a feature helps make good predictions.

AI pipeline and Intellectual Property:

AI Pipeline Intellectual property (IP) can be applied to various components of an AI pipeline, including algorithms, data, software code, and deployment strategies. Here are examples and solutions for applying IP to different stages of an AI.

Algorithms and Models:

Consider filing a patent for the unique aspects of your algorithm

Data Processing and Feature Engineering:

Document your data processing methods as trade secrets. Restrict access to these methods and consider patenting any novel preprocessing innovations.

Training Data and Software Code:

Clearly establish ownership of the dataset and use licensing agreements if you share it. Consider whether aspects of your dataset creation, such as collection methods, are patentable.

Copyright the code and include a clear license. If specific parts of the code are particularly innovative, consider filing for a patent.

Integration of Tools and Technologies:

Ensure compliance with open-source licenses. If the library is proprietary, make sure you have the necessary licenses. Protect your integration methods through patents or trade secrets.

Deployment and Monitoring:

Implement security measures to prevent unauthorized access to the deployed model.

Consider filing a patent for the optimization process. Document the process as a trade secret and restrict access to it to maintain a competitive advantage.

Documentation and Collaboration Agreements:

Maintain detailed records of your development process. This documentation can serve as evidence of your ownership and can be crucial in defending your IP rights.

Clearly define IP ownership and usage rights in collaboration agreements. Specify which parts of the project are joint efforts and establish how IP will be managed.

It's important to note that the application of IP to an AI pipeline can be complex and may involve a combination of patents, trade secrets, copyrights, and licenses. Consulting with intellectual property professionals, such as patent attorneys, can help you navigate the specific challenges and opportunities related to your AI project.