Overthinking on Overfitting in Machine Learning

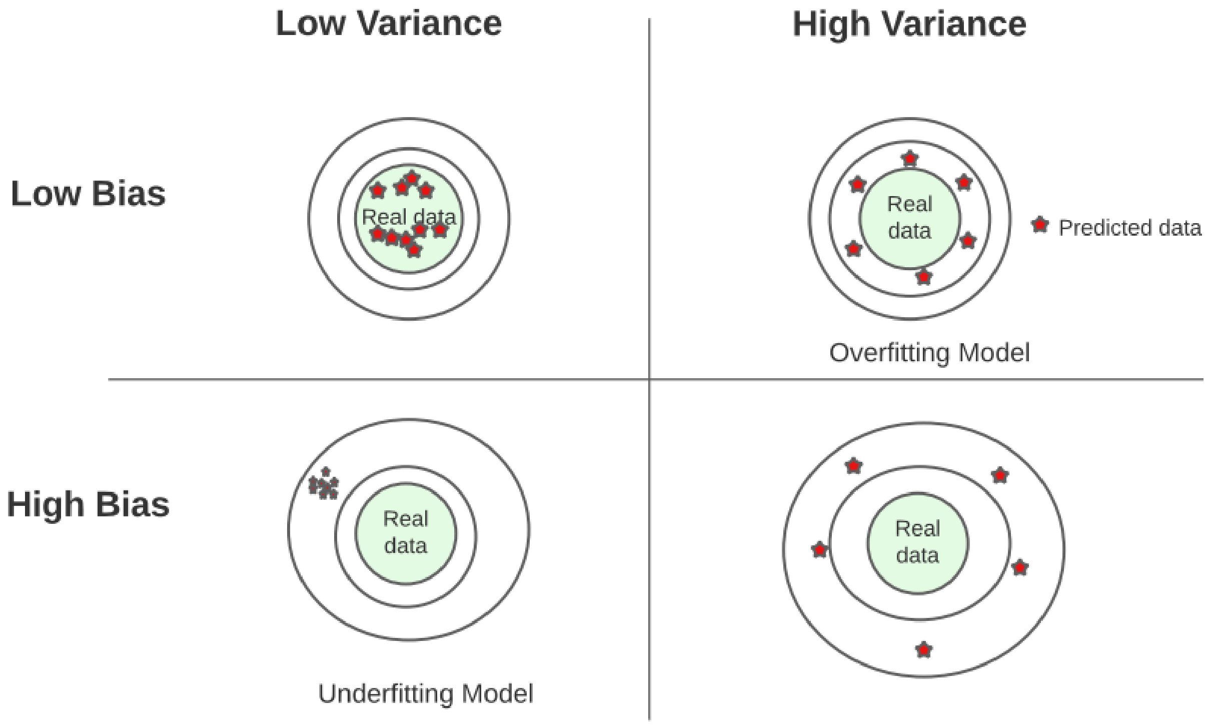
Sergey Zayats

One of the most common problems in machine learning is overfitting.

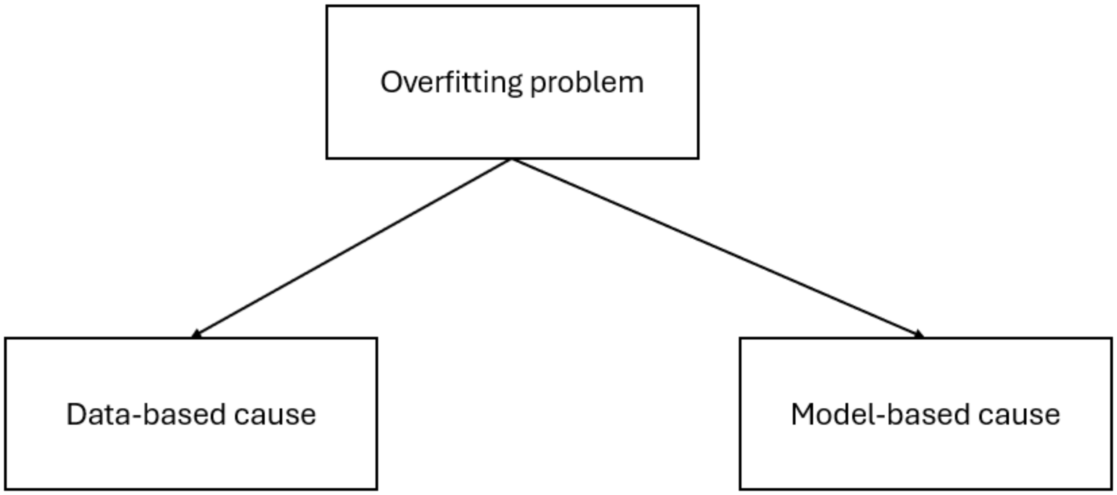
For those unfamiliar with the terminology, **overfitting** occurs when a model shows good results on the training data but performs poorly on the test (unseen) data.

Overfitting occurs when a model becomes highly sensitive to fluctuations in the training data, and it is characterized by:

- **Low Bias**: The model closely fits the training data, capturing its patterns and intricacies, leading to very accurate predictions on the training set.
- **High Variance**: The model fails to generalize to unseen data, as it has learned not only the underlying patterns but also the noise present in the training set. This sensitivity results in significantly poorer performance on the test set or new data.



Overfitting can be broadly categorized into two groups: **data-based** and **model-based** causes.



- **Data-based causes**: These arise from issues related to the dataset itself.
- **Model-based causes**: These stem from the model’s complexity and training process.

There is indeed a strong connection between the data and the model, and overfitting often occurs due to an interplay between the two. Each anti-overfitting technique tends to focus on one of these specific aspects.

At this point, people typically:

1. Start experimenting with the **model** (regularization, hyperparameter tuning, early stopping/pruning, changing architecture, etc.). This is often the easiest approach, as it simply involves tweaking tools and doesn't require knowledge beyond machine learning and data science.
2. Start experimenting with the **data** (using different cleaning approaches, augmentation, resampling, synthetic data generation, different validation strategies, etc.). Working with data usually presents more challenges and requires deeper understanding, but it’s the right direction.

What if we're trying to find a black cat in a dark room, especially when it isn’t there?

When we experiment on the training data, we do gain a better understanding of it. However, one critical point often doesn’t get enough attention: **data lineage**.

- **What is the origin of this data?**
- **How exactly was the data generated?**
- **Why was it generated in the way it is presented to us?**

At this stage, you require **external knowledge—**engaging with the people responsible for generating the data, those who oversee the sensors or systems that provide the data, etc. Once again, **Data and AI are not just about machines doing machine stuff**. Yes, to some extent they are, but ultimately, **Data and AI are fundamentally human endeavors**.

The success of any AI or machine learning model doesn't solely rely on algorithms or computational power. Instead, it depends on our understanding of the **context and quality** of the data we feed into these systems. This is why **data lineage** is so critical. It's not enough to just receive data and assume it's perfect. We need to trace it back to its origins, understand how it was collected, and critically assess why it looks the way it does.

We might be running experiments, tweaking models, and optimizing algorithms, but without a firm grasp of where the data came from or why it looks the way it does, we could be chasing nonexistent patterns or solving the wrong problems altogether.

To put it simply, an AI algorithm doesn't work like an ideal multicooker, where you can just throw in everything you have and expect to get a delicious, presentable dinner.

You need to be precise in selecting the right ingredients, using the right amounts, and determining what kind of additional preparation is needed for those ingredients. And if you take all the necessary steps correctly and thoroughly…

Bon appétit!