Source

Welcome to Week 6 Lecture 2!

Data Science in Python & Machine Learning

Announcements

1. <u>Belt Exam for Stack 2</u>
    a. March 11th - March 13th
    b. Must have attended 80% of classes
    c. Must have passed 90% of assignments
    d. All assignments & resubmits from weeks 1 & 2 are due by Friday at 9am PST (March 11th)

2. <u>Final Project</u>
    a. Have at least 5 slides with 2 visualizations.
    b. 1 slide should state the problem.
    c. 2 slides should include visualizations with analyses.
    d. 1 slide should include trends or insights you gained from the data.
    e. 1 slide should include recommendations for stakeholders.

Learning Goals

After this lesson you will be able to:

1. Explain bias, variance, and the bias/variance tradeoff.
2. Visualize how a Tree Based model makes predictions
3. Use regularization to reduce variance (overfitting) in a Tree Based model.

Introducing the Bias/Variance Tradeoff

Imagine you have a study guide for an upcoming test.

How are you going to use that study guide?

a) Memorize the study guide?
b) Focus on just on big ideas?

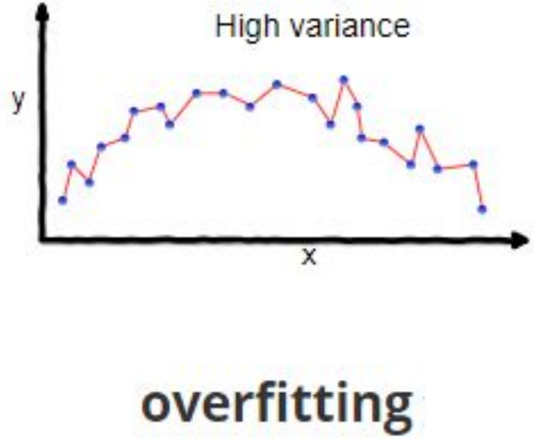
The effectiveness of your approach will only be known on TEST DAY!

High Variance = OVERfitting

Study approach A (memorize every detail) corresponds to a model with high variance.

You would do perfect on a test that was exactly the same as your study guide.

But what if there is a question not specifically addressed on your study guide? You have no ability to recognize big ideas to make a valid guess.



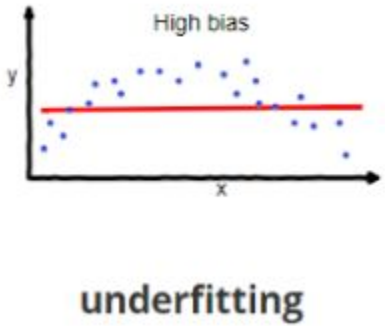
Notice how this model perfectly predicts each data point IF the "test" is the same as the "study guide".

High Bias = UNDERfitting

Approach B (only big ideas and no detail) corresponds to a model with high bias

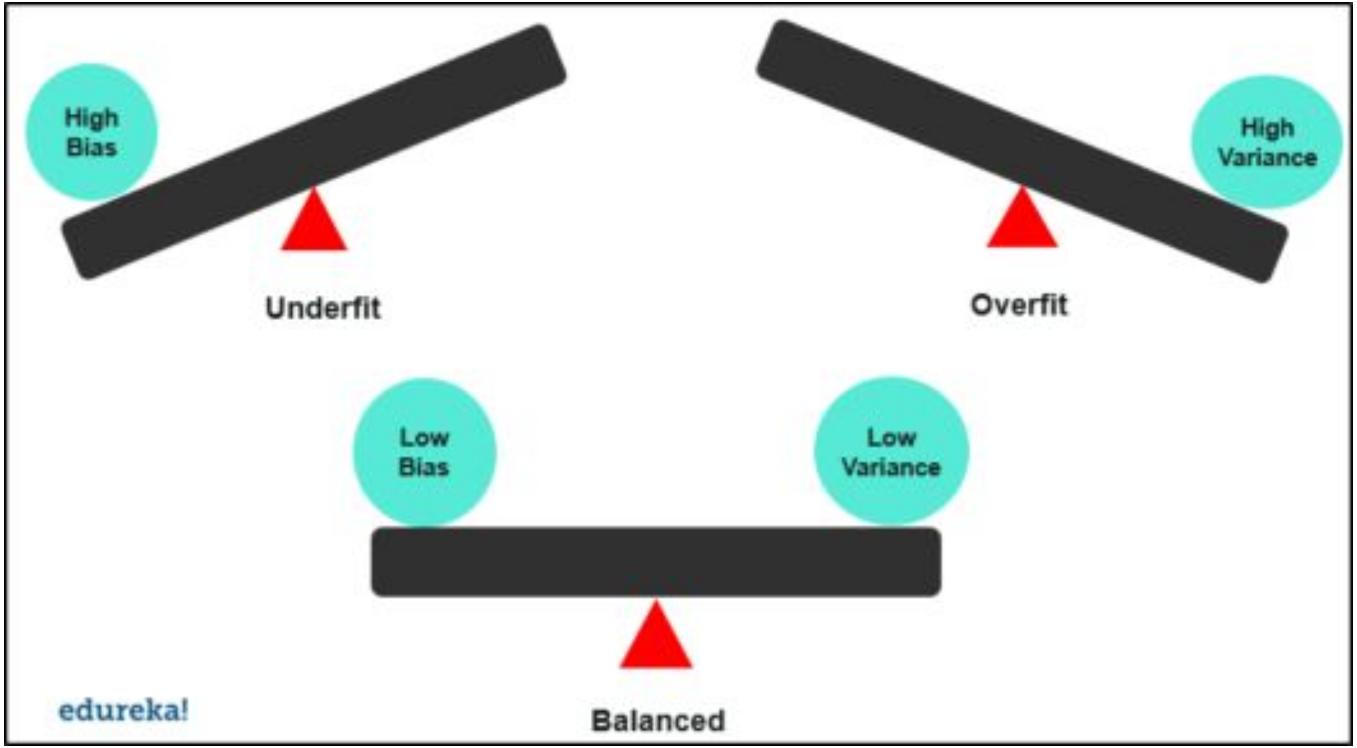
You would miss any question on the test that involved knowing details.

But what if there is a question from the study guide that goes into more detail? You could potentially miss this question because you did not focus on the details.



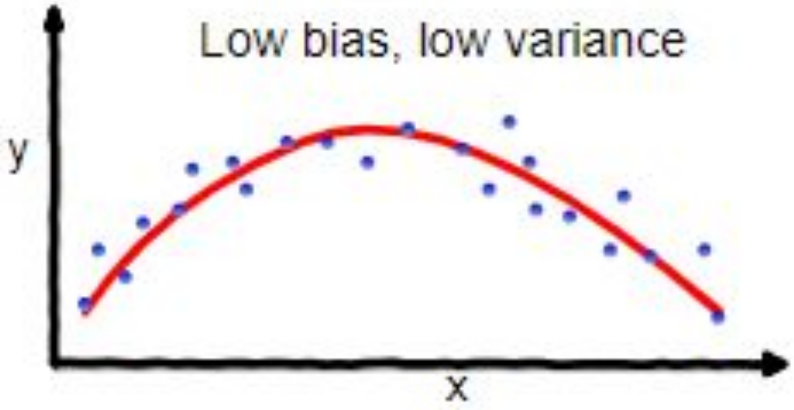
This model is in a reasonable range of the data.
Notice how this model is so simple that it misses details.

We want a balance!

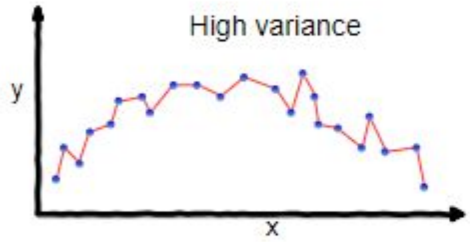


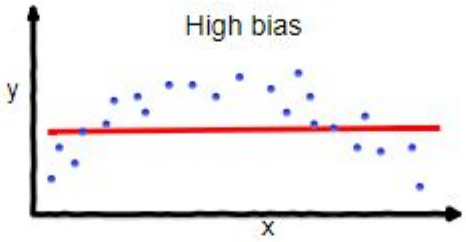
Source

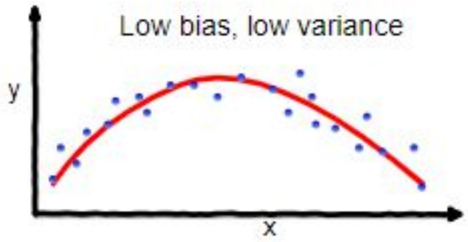
This is a nice balance between the big pattern and the details.
The model found the FUNCTION, but did not fit on the NOISE!



Good balance







overfitting

Model is too complex and fits on **random noise** in the data

underfitting

Model is too simple

Good balance

Model is just right and fits on the **general function**

Bias/Variance Tradeoff:

Decreasing the variance tends to increase the bias of the model

How to identify overfitting?

Train R2 Score	Test R2 Score	Bias/Variance
.30	.29	High Bias (underfit)

How to identify overfitting?

Train R2 Score	Test R2 Score	Bias/Variance
.30	.29	High Bias (underfit)
.99	.54	High Variance (overfit)

How to identify overfitting?

Train R2 Score	Test R2 Score	Bias/Variance
.30	.29	High Bias (underfit)
.99	.54	High Variance (overfit)
.45	.14	High Bias AND Variance

How to identify overfitting?

Train R2 Score	Test R2 Score	Bias/Variance
.30	.29	High Bias (underfit)
.99	.54	High Variance (overfit)
.45	.14	High Bias AND Variance
.95	.94	Good Fit

Regularization

Regularization prevents a model from overfitting.

We need to find the RIGHT amount of regularization to balance the bias and variance.

We call this 'tuning' a model

Image Source

Regularization

So your model has high variance, now what?

Regularization prevents a model from memorizing the data and forces it to find the general function instead.

- Done differently for different model types, for example:
    - Decision Trees: Adjust *max_depth, min_sample_split, min_sample_leaf* or other hyperparameters. Check the documentation for more options.
    - **Random Forests:** Adjust n_estimators, *max_depth,min_sample_split, min_sample_leaf* or other hyperparameters. Check the documentation for more options.
    - **K Nearest Neighbors:** Adjust n_neighbors, weights or other hyperparameters. Check the documentation for more options.

- Each model type has different ways to regularize it and combat overfitting. Check the documentation for each model type.

The Goal?

Maximize the Testing Metrics!!!

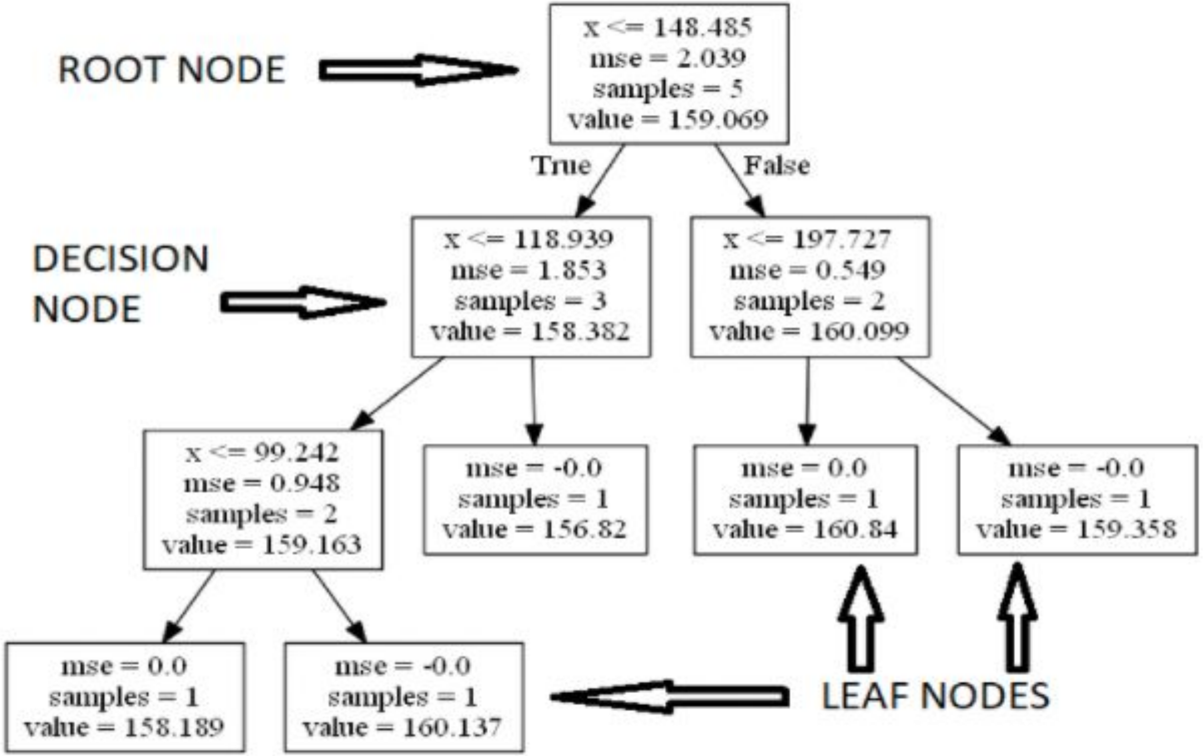
Don’t sacrifice test metrics to decrease variance

High variance means you might be able to do better on the test set if you use regularization.

Implementing Decision Trees

- A predictive modeling approach that separates data into classes using a top down approach
- Can be used for classification and regression
- Regression Trees predict a continuous quantity or numeric output such as $35.98 or 11.41 inches
- The sklearn.tree.DecisionTreeRegressor uses mean squared error, friedman mse and mean absolute error.

A Simple Decision Tree



Decision Tree Based Models

Advantages

- Easy to Interpret
- Prediction is fast
- Can be used for classification or regression
- Doesn’t require scaling
- Can be used for multiclass classification problems (more than 2 classes)

Disadvantages

- Worse performance than other supervised learning methods
- Prone to overfitting
- Small variations in the data can result in a completely different tree

```python
# Imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_selector, make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.dummy import DummyRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```python
cat_selector = make_column_selector(dtype_include='object')

ohe = OneHotEncoder(sparse=False, handle_unknown='ignore')

ohe_tuple = (ohe, cat_selector)

transformer = make_column_transformer(ohe_tuple, remainder='passthrough')
```

Decision Trees in Python

```
from sklearn.tree import DecisionTreeRegressor
```

```
dec_tree = DecisionTreeRegressor()

dec_tree_pipe = make_pipeline(transformer, dec_tree)

dec_tree_pipe.fit(X_train, y_train)
```

Bias/Variance Code Along

CodeAlong Notebook

Challenge Notebook