UNDERSTANDING BIAS VS VARIANCE TRADEOFF

IN PYTHON

DR. ALVIN ANG

Understanding Bias-Variance Tradeoff in Python

by Dr. Alvin Ang

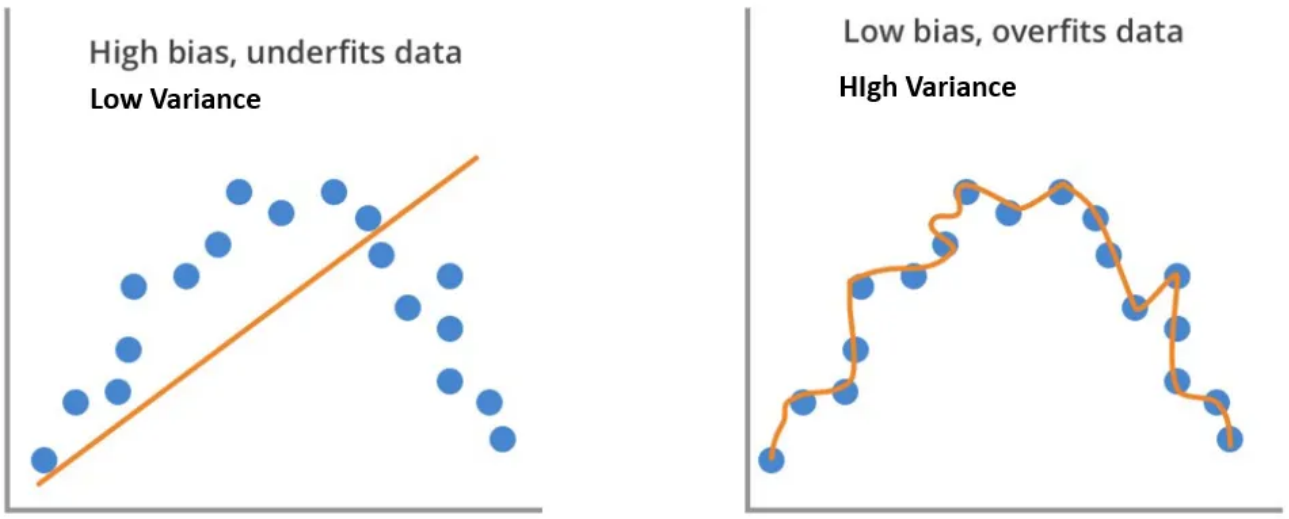
Dr. Alvin Ang

4 min read · 1 day ago

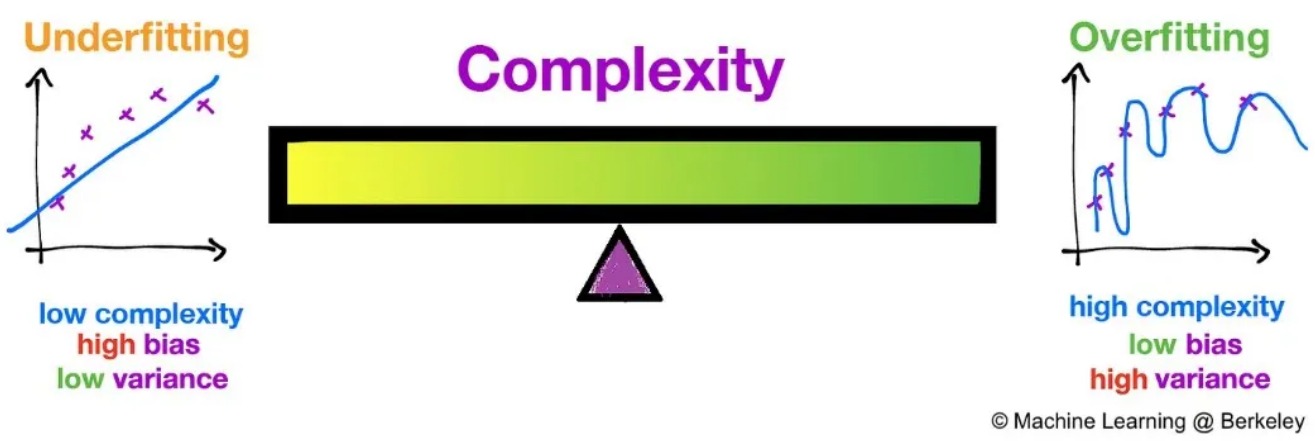
https://www.alvinang.sg/s/Understanding_Bias_vs_Variance_in_Python_by_Dr_Alvin_Ang.ipynb

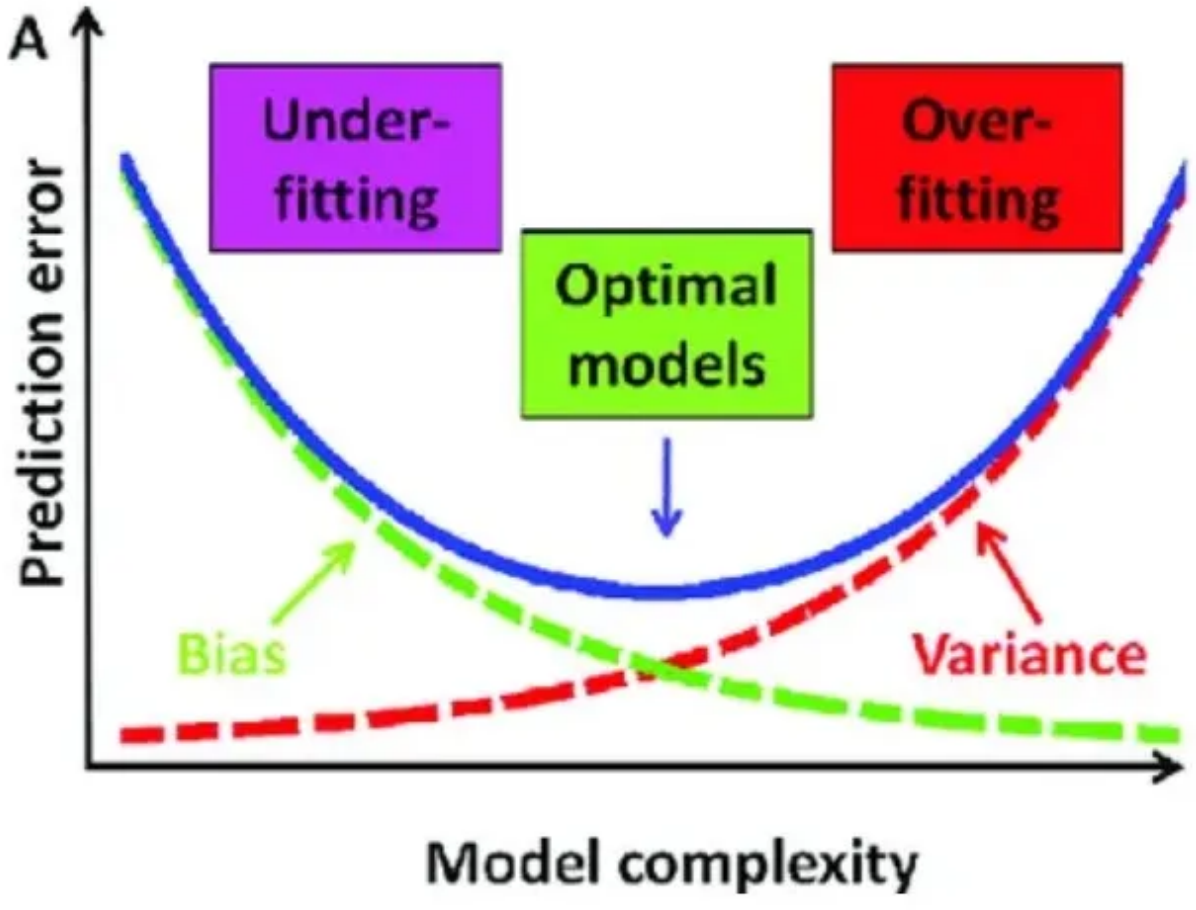
BIAS = UNDERFIT

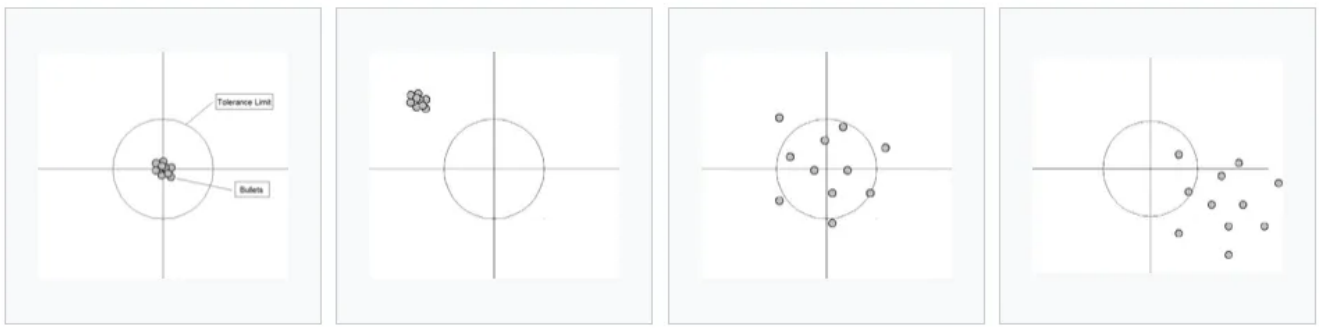
VARIANCE = OVERFIT



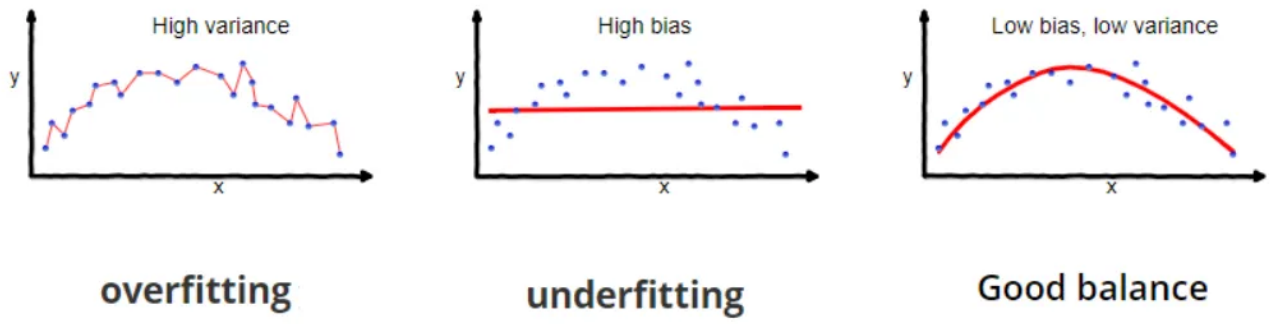
Bias of a simplistic (left) vs a complex model (right).

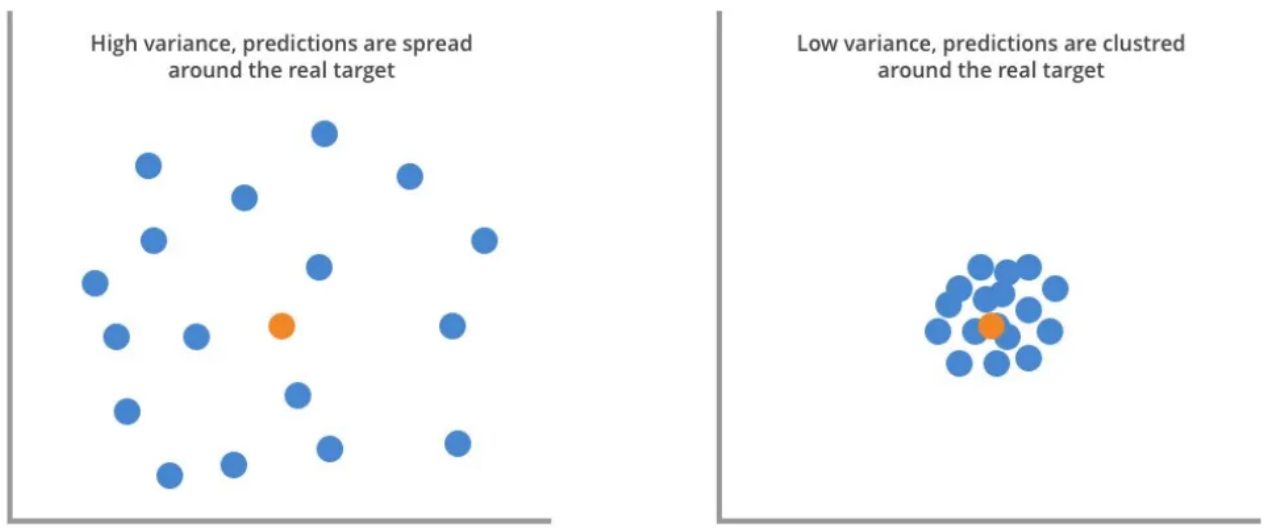




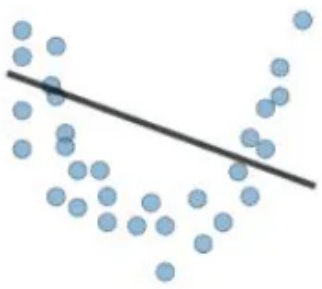
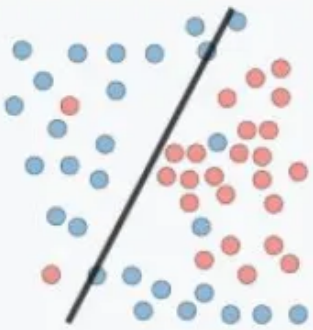
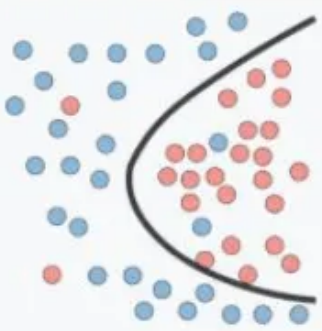
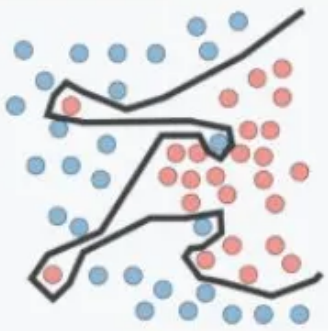


bias low, variance low | bias high, variance low | bias low, variance high | bias high, variance high





Range of predictions in a model with high (left) and low variance (right).

	Underfitting	Just right	Overfitting
Symptoms	• High training error • Training error close to test error • High bias	• Training error slightly lower than test error	• Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	• Complexify model • Add more features • Train longer		• Perform regularization • Get more data

How do we Measure Bias and Variance?

Answer: We use MSE (because it encompasses both Bias + Variance into ONE indicator)

$$MSE = bias^2 + variance$$

- A ***LOW MSE*** indicates that both the bias and the variance are low. This means that the model is **accurate and reliable**.
- A ***HIGH MSE*** indicates that either the bias or the variance is high. This means that the model is **not accurate nor reliable**.

> *In general,*

- MSE below 10 is considered to be low.
- 0 indicates that the model is perfect and is making no errors.
- 1 indicates that the model is making as many errors as it is making correct predictions.
- MSE between 10 and 100 is considered to be high.
- MSE above 100 is considered to be very high. It is VERY inaccurate.

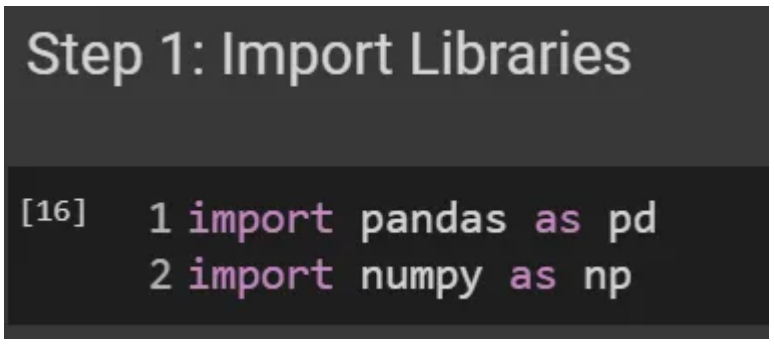
The definition of high or low for the MSE depends on the specific application.

However, it is important to note that these are just guidelines. The definition of high or low for the MSE will vary depending on the specific application.

For example, a MSE of 10 might be considered to be low for a model that is predicting the price of a house, but it might be considered to be high for a model that is predicting whether or not a patient will have a heart attack.

How to Obtain the MSE of the Model using Python?

https://www.alvinang.sg/s/Understanding_Bias_vs_Variance_in_Python_by_Dr_Alvin_Ang.ipynb



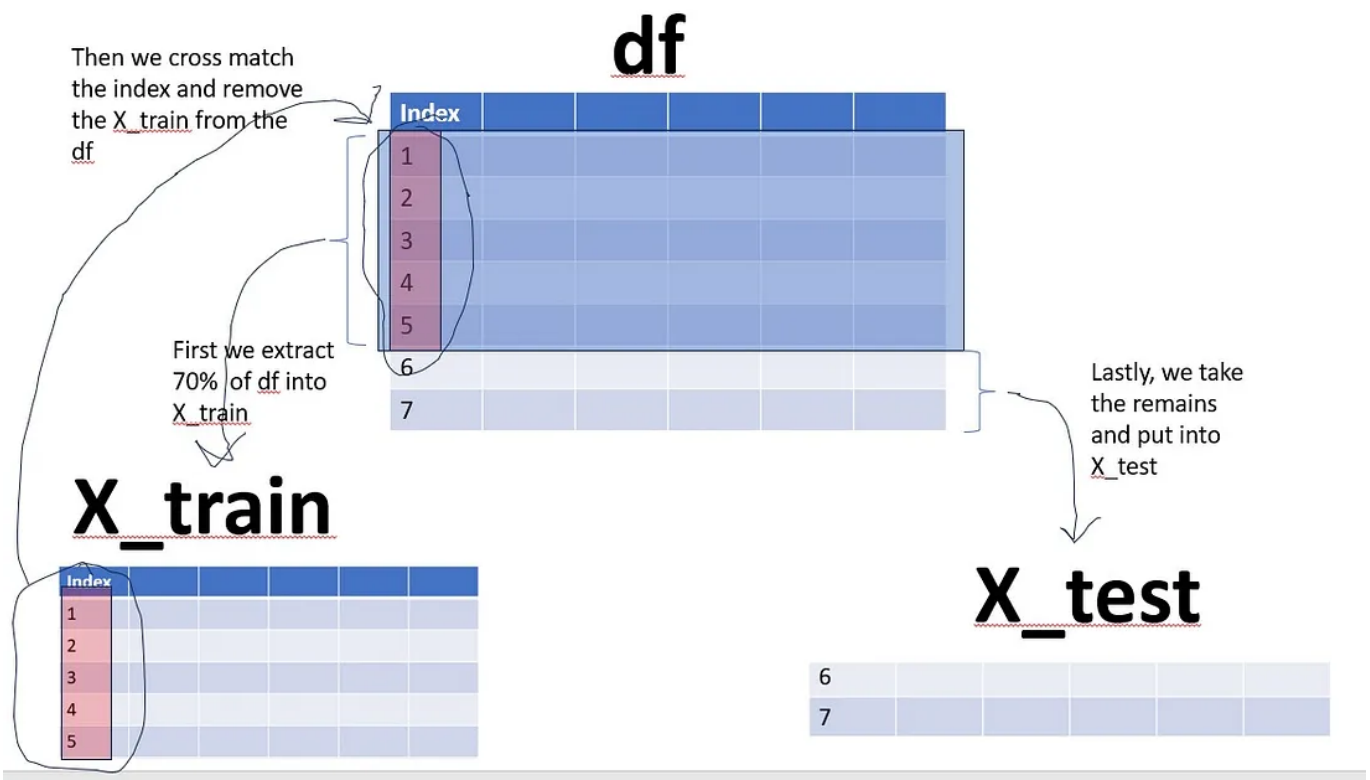
Step 2: Load the Boston Housing Data.csv

```
1 # Load the Boston housing dataset.
2 df = pd.read_csv('https://www.alvinang.sg/s/boston_housing_data.csv')
3 df
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.64	23.9

Step 3: Train Test Split

```
[18] 1 #X_train will sample 70% of the df
     2 X_train = df.sample(frac=0.7)
     3
     4 #then, we drop off the X_train from the df (since its a subset of it)
     5 #this is done by matching the similar index column rows via the .index
     6 #then, we keep the remaining and store it into X_test
     7
     8 X_test = df.drop(X_train.index)
```




```
1 y_train = X_train.pop('MEDV')
2
3 y_test = X_test.pop('MEDV')
```

Step 4: Create the LR Model using SKLearn

```
[20] 1 from sklearn.linear_model import LinearRegression
     2
     3 model = LinearRegression()
```

Step 5: Train / Fit the model using Linear Regression (using X_train and y_train)

```
[21] 1 # Train the model on the training data.
     2 model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

Step 6: Use the Model on the X_test to Predict the y_pred

```
1 # Predict the values of y using the model.
2 y_pred = model.predict(X_test)
```

Step 7: Find the MSE

```
[23] 1 import sklearn.metrics
     2
     3 mse = sklearn.metrics.mean_squared_error(y_test, y_pred)
     4
     5 print("The mean squared error is:", mse)
```

```
The mean squared error is: 24.90477097634145
```

THE END

References:

Calculation of Bias & Variance in python

Bias-Variance Decomposition Demystified

medium.com

Holy Grail for Bias-Variance tradeoff, Overfitting & Underfitting

Explained unconventionally, it will be worth your time, read on:)

juhiramzai.medium.com

About Dr. Alvin Ang

www.AlvinAng.sg

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. Previously he was a Principal Consultant (Data Science) as well as an Assistant Professor. He was also 8 years SUSS adjunct lecturer. His focus and interest is in the area of real world data science. Though an operational researcher by study, his passion for practical applications outweigh his academic background He is a scientist, entrepreneur, as well as a personal/business advisor.

More about him at www.AlvinAng.sg.

Python | Bias Variance Tradeoff | Bias | Variance | Bias Variance Trade Off

Written by Dr. Alvin Ang

Edit profile

67 Followers

www.AlvinAng.sg | www.linkedin.com/in/dr-alvin | www.datafrens.sg | https://public.tableau.com/app/profile/dr.alvin.ang