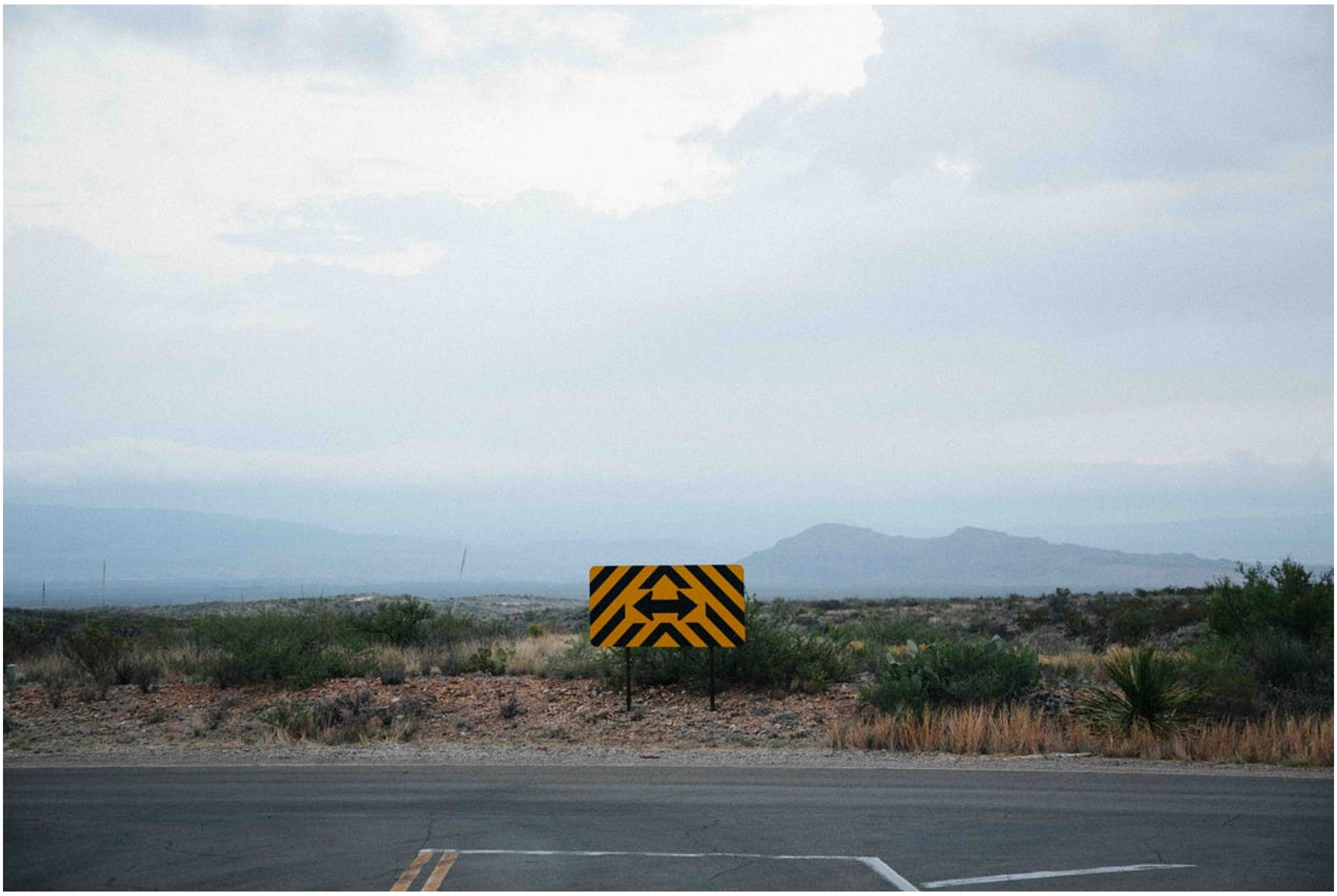
Akaike Information Criterion: Model Selection

Adi tya

Aditya Manikantan · Follow

Published in Geek Culture

6 min read · Jun 10, 2021



Akaike Information Criterion or AIC is a statistical method used for **model selection.** It helps you compare **candidate models** and select the best among them.

Candidate models can be models each containing a different subset or combination of independent/predictor variables.

AIC aims to select the model which **best** explains the variance in the dependent variable with the **fewest** number of independent variables (parameters). So it helps select a **simpler** model (fewer parameters) over a **complex** model (more parameters).

But why select a simpler model over a complex one?

- To reduce overfitting:

We know that the more complex the model, the better it fits. However, this increase in complexity could lead to overfitting i.e low bias (high train accuracy) and high variance (low test accuracy). Therefore, AIC helps deal with this trade-off between a simple and a complex model.

- Reduce the number of parameters (reduce in the number of dimensions):

There is an added computational cost associated with adding a parameter. Also, unwanted parameters could result in the addition of noise which hinders the **goodness-of-fit** in the model. AIC score helps determine whether the cost of adding any given parameter is justified.

AIC measures the information lost, so the model with a **lower** AIC score indicates a **better fit**.

AIC is comprised of two important aspects

- Maximum log-likelihood (measures how well the given model as captured the variance in the dependent variable)
- Number of parameters

It's calculated using the formula:

$$\mathrm{AIC} = N * ln\left(\frac{SS_e}{N}\right) + 2K$$

$\boldsymbol{N}$: $Number\ of\ observations$
$\boldsymbol{SS_e}$: $Sum\ square\ of\ errors$
$\boldsymbol{K}$: $Number\ of\ parameters$

Since a smaller AIC score is preferred, based on this formula adding more parameters actually **penalizes** the score. So if two models equally explain the variance in the given data, the model with fewer parameters will have a lower AIC score and will be selected as the better fit model.

When is AIC required?

- Suppose for a given problem statement you have collected or scraped the necessary variables using your domain knowledge, but you're not sure whether these are important indicators for the problem.
- You lack the required amount of data to properly test the accuracy.

An important point to note is that the AIC score on its own has no significance. It has to be **compared** with another model.

Let's dip deeper using an example

Suppose I have a regression problem where I have to predict the price of a car. Let me give you an overview of the dataframe.

```
df.head()
```

	horsepower	enginesize	highwaympg	price
0	111	130	27	13495.0
1	111	130	27	16500.0
2	154	152	26	16500.0
3	102	109	30	13950.0
4	115	136	22	17450.0

- Independent variables: ***horsepower, engine size, highway mpg***
- Dependent variable: ***price***

There 3 parameters. So,

- K = 3 + 1 = 4 (Number of parameters in the model + Intercept)

Therefore, the number of subsets (combinations of given parameters) is 2^number of parameters = $2^3 = 8$, so in other words, there are 8 candidate models.

```
# Print the subsets of parameters
import itertools

for i in range(len(all_cols)+1):
    for subset in itertools.combinations(all_cols, i):
        print(list(subset))
```

```
[]
['horsepower']
['enginesize']
['highwaympg']
['horsepower', 'enginesize']
['horsepower', 'highwaympg']
['enginesize', 'highwaympg']
['horsepower', 'enginesize', 'highwaympg']
```

Here the empty set refers to an intercept-only model, the simplest model possible.

I'll be using Linear Regression to fit the given models.

```
y = df['price']
r2_scores = []
predictor_subsets = []

for i in range(len(all_cols)+1):
    for subset in itertools.combinations(all_cols, i):
        model = LinearRegression(n_jobs = -1, normalize=False)

        cols = list(subset)
        predictor_subsets.append(cols)

        # If intercept-only model
        if len(cols) < 1:
            x = np.full(len(y), 0)
            x = x.reshape(-1, 1)

            model.fit(x, y)

            ypred = model.predict(x

            score = model.score(x, y)
            r2_scores.append(score)
```

```
        else:
            x = df[cols]

            model.fit(x, y)

            ypred = model.predict(x)

            score = model.score(x, y)
            r2_scores.append(score)
```

These are the R2 scores after fitting each model:

```
results_df = pd.DataFrame({'Predictor Subset': predictor_subsets,
                           'R2 Score': r2_scores})
```

	Predictor Subset	R2 Score
7	[horsepower, enginesize, highwaympg]	0.797512
4	[horsepower, enginesize]	0.793341
6	[enginesize, highwaympg]	0.784659
2	[enginesize]	0.764129
5	[horsepower, highwaympg]	0.666894
1	[horsepower]	0.653088
3	[highwaympg]	0.486644
0	[]	0.000000

You can see that the top-scoring model consists of all the parameters whereas the second model contains all except *highwaympg*, but the difference in their R2 score is quite trivial. So is this slight increase in the R2 score justified?

To find out let's first calculate the AIC score for each candidate model.

```
# Function to calculate the AIC score
# N: number of obervations
# K: Number of parameters
# mse: Mean squared error (SSe/N)

def calculate_aic(N, mse, K):
    aic = N*np.log(mse)+2*K
    return aic
```

```
y = df['price']
aic_scores = []

for i in range(len(all_cols)+1):
    for subset in itertools.combinations(all_cols, i):
        model = LinearRegression(n_jobs = -1, normalize=False)

        cols = list(subset)

        #If intercept-only model
        if len(cols) < 1:
            x = np.full(len(y), 0)
            x = x.reshape(-1, 1)

            model.fit(x, y)

            ypred = model.predict(x)

            N = len(y)
            K = len(model.coef_) + 1
            mse = mean_squared_error(y, ypred)

            aic = calculate_aic(N, mse, K)
            aic_scores.append(aic)

        else:
            x = df[cols]

            model.fit(x, y)

            ypred = model.predict(x)

            N = len(y)
            K = len(model.coef_) + 1
            mse = mean_squared_error(y, ypred)

            aic = calculate_aic(N, mse, K)
            aic_scores.append(aic)
```

The AIC scores are:

	Predictor Subset	AIC score
7	[horsepower, enginesize, highwaympg]	3363.776559
4	[horsepower, enginesize]	3365.956282
6	[enginesize, highwaympg]	3374.392079
2	[enginesize]	3391.060017
5	[horsepower, highwaympg]	3463.820853
1	[horsepower]	3470.146075
3	[highwaympg]	3550.485264
0	[]	3687.176533

As you can see the AIC score of the **best model** (model with the **lowest** AIC score) is only slightly lower than the second-best model. For the extra parameter to be justified, the AIC score has to be lower by at least **2 units.**

Let's calculate **Delta AIC** for each model. Delta AIC is just the **difference** of the AIC score of each model from the best model. So, the Delta AIC of the best model should be 0.

```
results_df['Delta AIC'] = results_df['AIC score']-
min(results_df['AIC score'])
```

	Predictor Subset	AIC score	Delta AIC
7	[horsepower, enginesize, highwaympg]	3363.776559	0.000000
4	[horsepower, enginesize]	3365.956282	2.179723
6	[enginesize, highwaympg]	3374.392079	10.615520
2	[enginesize]	3391.060017	27.283458
5	[horsepower, highwaympg]	3463.820853	100.044294
1	[horsepower]	3470.146075	106.369516
3	[highwaympg]	3550.485264	186.708705
0	[]	3687.176533	323.399974

You can see that the AIC score of the best model is more than 2 units lower than the second-best model. Since the difference in the AIC scores is significant enough, we can conclude that the slight increase in R2 score by adding *highwaympg* is justified.

In other words, the increase in the variance explained by adding *highwaympg* is crucial enough for it to be added.

We can go a step further by calculating the **weighted AIC score** for each model. The weighted AIC score gives the **predictive power** of a given model with respect to all the other models.

To calculate weighted AIC first, calculate the **relative likelihood** of the model which is just exp(-0.5 * Delta AIC) of a model divided by the sum total of weighted AIC scores of all models.

```
results_df['Weighted AIC'] = round(np.exp(-0.5 * results_df['Delta AIC'])/sum(np.exp(-0.5 * results_df['Delta AIC'])), 4)
```

	Predictor Subset	AIC score	Delta AIC	Weighted AIC	R2 Score
7	[horsepower, enginesize, highwaympg]	3363.776559	0.000000	0.7456	0.797512
4	[horsepower, enginesize]	3365.956282	2.179723	0.2507	0.793341
6	[enginesize, highwaympg]	3374.392079	10.615520	0.0037	0.784659
2	[enginesize]	3391.060017	27.283458	0.0000	0.764129
5	[horsepower, highwaympg]	3463.820853	100.044294	0.0000	0.666894
1	[horsepower]	3470.146075	106.369516	0.0000	0.653088
3	[highwaympg]	3550.485264	186.708705	0.0000	0.486644
0	[]	3687.176533	323.399974	0.0000	0.000000

This table illustrates that the top 2 models explain almost 100% of the variance when compared to all the candidate models.

So the best model is the candidate model which includes all the independent variables in the dataframe. It has the lowest AIC score and contains about 75% of predictive power compared to the 25% by the second-best model.

Based on the above analysis, you can choose the given best model consisting of all independent variables to predict the *price* of the cars.

Summary

- Akaike Information Criterion helps you compare and select the best candidate model.
- The model with a lower AIC score shows a better fit.
- Prefers model which explains most variance with least parameters.
- Penalizes models with more parameters.
- AIC score has to be at least 2 units lower compared to the other model for it to be significant enough.
- Weighted AIC shows the predictive power of a given model with respect to other models.
- AIC score on its own has no significance. It has to be compared with another model.

Data: Source

Written by Aditya Manikantan

41 Followers · Writer for Geek Culture

Hey! I am interested in AI, Technology, Statistics, and how AI could impact well-being.