

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



# Hiperparametre Optimizasyonu

GridSearch + Cross-Validation



Merve Bayram Durna · [Follow](#)

Published in Bilişim Hareketi

5 min read · Jun 24, 2020



Share



More



Photo by [Will Francis](#) on [Unsplash](#)

Bir önceki yazımda cross-validation yönteminden ve bu yöntemin model

[Open in app](#)



Search



Cross-validation mükemmel bir kullanım alanı olan hiperparametre optimizasyonuna aşağıdaki alt başlıklar çerçevesinde ele alacağım.

1. Hiperparametre Nedir?
2. Hiperparametre Optimizasyonu Nedir?
3. Hiperparametre Optimizasyonu Yöntemleri Nelerdir?

### 3.1. GridSearchCV

### 3.2. RandomizedCV

## 4. Sklearn ile Hiperparametre Optimizasyonu Uygulaması

### 1. Hiperparametre Nedir ?

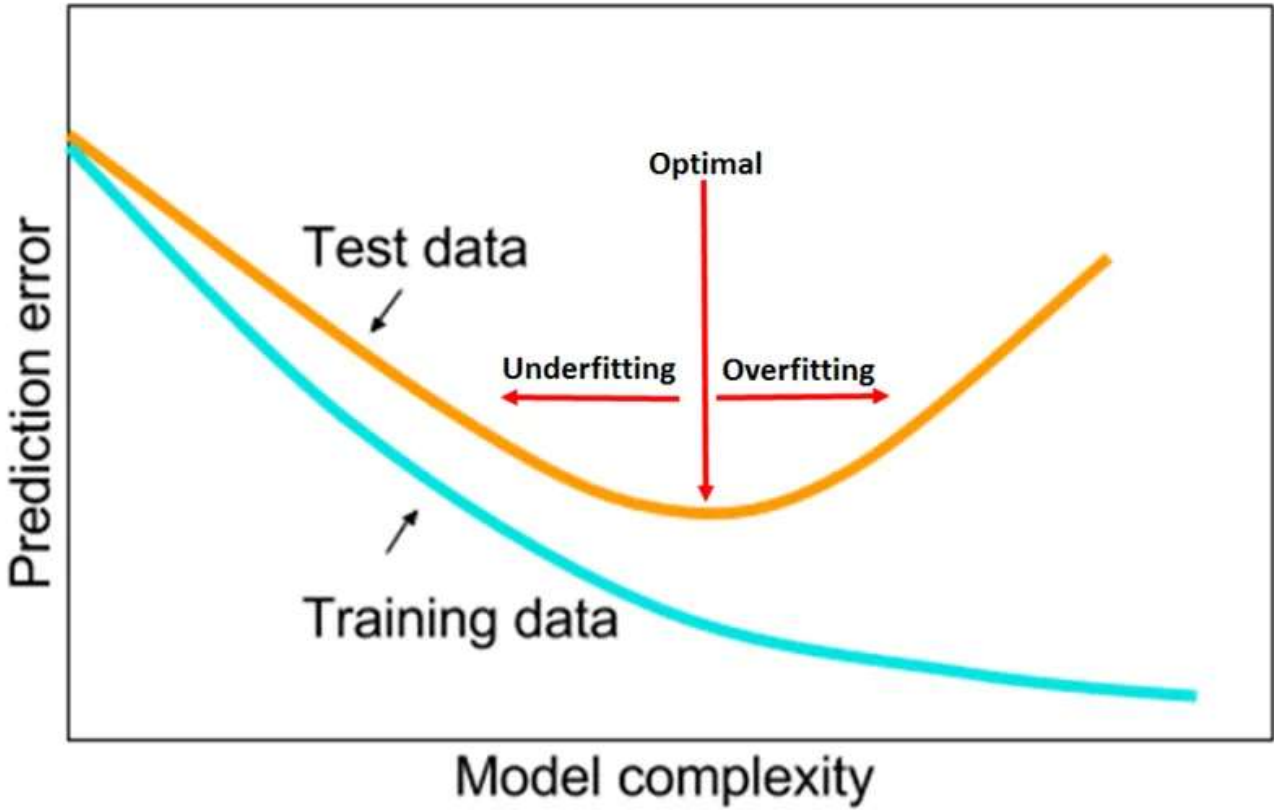
Makine öğrenmesi modelleri temel olarak bağımlı ve bağımsız değişkenler (target-features) arasındaki ilişkiyi temsil eden matematiksel fonksiyonlara dayanır. Örneğin Lineer Regresyon bağımlı ve bağımsız değişkenler arasında  $y = mx + b$  fonksiyonuyla ifade edilebilecek doğrusal bir ilişki olduğunu varsayar. Lineer Regresyon modelinin eğitilmesi sırasında çeşitli optimizasyon teknikleri kullanılarak en uygun m ve b değerleri belirlenir bu değerler model parametreleridir.

Parametrelerden farklı olarak **hiperparametreler** modelin eğitilmesi sırasında öğrenilmez. Modelleme aşamasının öncesinde veri bilimci tarafından belirlenirler. Örneğin non-parametrik sınıflandırma algoritmalarından olan KNN algoritması tahmin edilmek istenen değere en yakın k tane komşusuna bakarak sınıflandırma yapar. Burada k sayısı (**n\_neighbors** : ) ve kullanılacak olan uzaklık metriği (**metric** : ) modelleme öncesinde veri bilimci tarafından belirtilmesi gereken, modelin performansını arttıran hiperparametrelerdendir.

### 2. Hiperparametre Optimizasyonu Nedir?

Hiperparametre optimizasyonu, bir makine öğrenmesi algoritması için belirlenen başarı metriğine göre en uygun hiperparametre kombinasyonunu bulma işlemidir.

- Hiperparametre optimizasyonu ile model karmaşıklığı dengelenerek overfitting ve underfitting dengesi sağlanabilir.
- Yine modelin esnekliğinden kaynaklanan aşırı öğrenme problemi hiperparametreler ile getirilen kısıtlamalarla çözülebilir.



- Hiperparametre optimizasyonu için iyi bir temel(base) modele ihtiyaç vardır. Başarısız bir base model üzerinde hiperparametre optimizasyonunun harikalar yaratmasını bekleyemeyiz. Bu aşamada geri dönüp veri ön işleme ve özellik mühendisliği kısımlarında daha fazla zaman harcayarak, iyi bir base model kurduktan sonra hiperparametre optimizasyonu yapmak daha iyi sonuçlar elde edilmesini sağlayacaktır.

### 3. Hiperparametre Optimizasyonu Yöntemleri

Bir makine öğrenmesi algoritması için onlarca hiperparametre ve bu hiperparametrelerin alabileceği onlarca değer olduğu düşünüldüğünde, tüm kombinasyonları tek tek denemenin ve en iyi kombinasyonu seçmenin ne kadar zor olacağı açıktır. Bu sebeple hiperparametre optimizasyonu için farklı yöntemler geliştirilmiştir. GridSearchCV ve RandomizedSearchCV bu yöntemlerdendir. Şimdi de bu iki yöntemin çalışma prensibine, artı ve eksi yönlerine bakalım :

#### 3.1. GridSearchCV

- Modelde denenmesi istenen hiperparametreler ve değerleri için bütün kombinasyonlar ile ayrı ayrı model kurulur ve belirtilen metriğe göre en başarılı hiperparametre seti belirlenir.
- (+) Tüm kombinasyonları denendiği için en iyi performans gösteren hiperparametre setini belirlemeyi garanti eder. Küçük veri setlerinde ve sadece

birkaç tane hiperparametre denenmek istendiğinde çok iyi çalışır.

- (-) Büyük bir veri seti ile çalışıldığında ya da denenecek olan hiperparametre sayısı ve değeri arttırıldığında kombinasyon sayısı da katlanarak artacaktır. Kurulan her modelin cross-validation ile test edildiği de düşünüldüğünde maliyet korkunç derecede artacaktır bu sebeple alternatif olarak RandomSearchCV yöntemi tercih edilebilir.

### 3.2. RandomizedSearchCV

- Rastgele olarak bir hiperparametre seti seçilir ve cross-validation ile model kurularak test edilir. Belirlenen hesaplama süresi limitine ya da iterasyon sayısına ulaşıncaya kadar bu adımlar devam eder.
- (+) Büyük veri setlerinde daha az maliyetle GridSearchCV yöntemiyle elde edilen en iyi skora yakın performans gösterecek hiperparametre setlerini belirleyebilir.
- (+) Daha geniş bir hiperparametre alanını tarayabilir.
- (-) Her ne kadar optimum hiperparametre setine yaklaşılsa da tüm olası kombinasyonları tek tek denemediği için en iyi performans gösteren hiperparametre setini bulmayı garanti edemez.

Maaliyet açısından düşünüldüğünde gerçek hayatta kendine GridSearchCV den daha fazla yer bulur.

### 4. Sklearn ile Hiperparametre Optimizasyonu Uygulaması

Bu bölümde sklearn kütüphanesinden import edeceğimiz “wine” veri seti üzerinden RandomForest algoritmasını kullanarak hiperparametre optimizasyonu yapacağız.

Uygulama sırasında kullanacağımız kütüphaneleri ve veri setini import ederek başlayalım.

Veri setimiz yüklendi, train-test seti olarak ikiye bölündü şimdi modelleme için hazır. İlk olarak hiperparametre belirtmeden RandomForest algoritması ile base model oluşturacağız ardından manual bir şekilde hiperparametre optimizasyonu yapmayı son olarak da GridSearchCV ve RandomizedSearchCV yöntemlerini ele alacağız. Elde ettiğimiz accuracy skorlarını karşılaştırarak yöntemlerin artı ve eksi yanlarını uygulama üzerinde gözlemleyeceğiz.

Not : Uygulama kapsamında veri manipulasyonu, veri ön işleme ve özellik mühendisliği gibi kısımlar ele alınmayacaktır. Base model skorunu geliştirmek için bu kısımlar ele alınıp üzerinde zaman geçirilmeli ve sonrasında hiperparametre optimizasyonu yapılmalıdır. Yukarıda da belirttiğimiz gibi kötü bir base model üzerinden hiperparametre optimizasyonu ile harikalar yaratmayı bekleyemeyiz.

## Base Model

Base Model

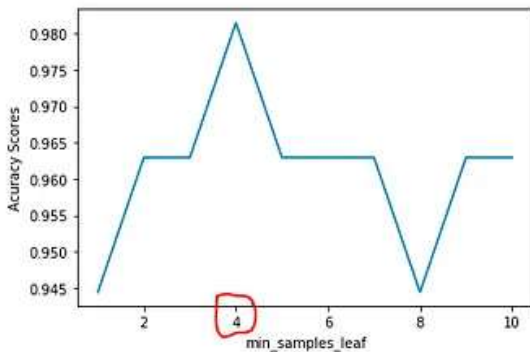
```
rf base model accuracy score = 0.9629629629629629
```

Veri setimiz çok karmaşık ve büyük olmadığından veri ön işleme kısımlarını ele almamamıza rağmen iyi bir accuracy skoru elde ettik. Şimdi de hiperparametre optimizasyonu ile bu skoru daha da arttırmaya çalışalım.

## Manual Hiperparametre Optimizasyonu

Bu bölümde amacımız Random Forest algoritmasının hiperparametrelerinden olan “min\_samples\_leaf” için farklı değerler deneyerek accuracy skorlarının nasıl değiştiğini gözlemlemek.

[0.9444444444444444, 0.9629629629629629, 0.9629629629629629, 0.9814814814814815, 0.9629629629629629, 0.9629629629629629, 0.9629629629629629, 0.9444444444444444, 0.9629629629629629, 0.9629629629629629]



Manual optimizasyon sonucu elde ettiğimiz çıktı ve görseli incelediğimizde `min_samples_leaf=4` için modelin en yüksek accuracy skorunu verdiğini ve bu skorun 0.981 olduğunu gözlemledik.

Burada sadece bir hiperparametre üzerinden 10 farklı değeri deneyerek en iyi sonucu veren hiperparametre değerini bulmaya çalıştık. Denenmek istenen çok sayıda hiperparametre ve değeri olduğunda optimizasyonun manual olarak yapılamayacağı açıktır. Bu ihtiyaçtan dolayı GridSearchCV ve RandomizedSearchCV yöntemleri geliştirilmiştir. Şimdi de bu yöntemlerin nasıl uygulanacağına bakalım.

## GridSearchCV İle Hiperparametre Optimizasyonu

İlk olarak denenmesi istenen hiperparametreler ve değerleri bir sözlük yapısında tanımlanır. Daha sonra Sklearn kütüphanesinden import edilen GridSearchCV metodu çağırılıp gerekli parametreleri belirtilir.



```
Fitting 5 folds for each of 1200 candidates, totalling 6000 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.  
[Parallel(n_jobs=-1)]: Done 33 tasks      | elapsed: 6.7s  
[Parallel(n_jobs=-1)]: Done 154 tasks     | elapsed: 54.1s  
[Parallel(n_jobs=-1)]: Done 357 tasks     | elapsed: 2.0min  
[Parallel(n_jobs=-1)]: Done 640 tasks     | elapsed: 3.5min  
[Parallel(n_jobs=-1)]: Done 1005 tasks    | elapsed: 5.5min  
[Parallel(n_jobs=-1)]: Done 1450 tasks    | elapsed: 8.3min  
[Parallel(n_jobs=-1)]: Done 1977 tasks    | elapsed: 13.2min  
[Parallel(n_jobs=-1)]: Done 2584 tasks    | elapsed: 20.2min  
[Parallel(n_jobs=-1)]: Done 3273 tasks    | elapsed: 31.5min  
[Parallel(n_jobs=-1)]: Done 4042 tasks    | elapsed: 39.0min  
[Parallel(n_jobs=-1)]: Done 4893 tasks    | elapsed: 47.0min  
[Parallel(n_jobs=-1)]: Done 5824 tasks    | elapsed: 55.8min  
[Parallel(n_jobs=-1)]: Done 6000 out of 6000 | elapsed: 57.6min finished
```

```
rf_gridcv_model.accuracy_score = 0.992
```

Yukarıdaki kodun çıktısında; 6 tane hiperparametre için 1200 tane farklı kombinasyon olduğunu, her bir kombinasyona 5-katlı cross-validation

uygulandığında toplamda 6000 tane model fit etme işleminin gerçekleştiğini ve bu işlemlerin 57.6 dakika sürdüğünü görebiliyoruz.

Optimizasyon sonucunda belirlenen en iyi performansı gösteren hiperparametre değerleri kullanıldığında accuracy skoru 0.992 oluyor.

Sonuç gayet iyi olsada, küçük bir veri seti olmasına rağmen tüm bu işlemlerin yapılması neredeyse 1 saat sürdü. Zaman açısından değerlendirildiğinde çok maliyetli olduğu açıktır. Bu sebeple RandomizedCV yöntemi geliştirilmiştir. Tüm ihtimalleri tek tek denemediği için her zaman en iyi skoru vereceğini garanti edemesede, maliyet açısından değerlendirildiğinde pratikte kendisine GridSearchCV den daha fazla yer buluyor. Şimdi de RandomizedSearchCV yöntemini ele alalım.

### **RandomizedSearchCV ile Hiperparametre Optimizasyonu**

Burada da GridSearchCV ye benzer şekilde, denenmesi istenen hiperparametreler ve değerleri bir sözlük yapısında tanımlanır. Daha sonra Sklearn kütüphanesinden import edilen RandomizedSearchCV metodu çağırılıp gerekli parametreleri belirtilir.

Bu yöntemle 1200 tane kombinasyonu ayrı ayrı denemek yerine n\_iter= 200 belirleyerek 200 farklı kombinasyonunu deneyeceğiz.

```
Fitting 5 folds for each of 200 candidates, totalling 1000 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.  
[Parallel(n_jobs=-1)]: Done 33 tasks      | elapsed: 12.0s  
[Parallel(n_jobs=-1)]: Done 154 tasks    | elapsed: 1.0min  
[Parallel(n_jobs=-1)]: Done 357 tasks    | elapsed: 2.6min  
[Parallel(n_jobs=-1)]: Done 640 tasks    | elapsed: 4.4min  
[Parallel(n_jobs=-1)]: Done 1000 out of 1000 | elapsed: 6.5min finished
```

```
rf randomcv model accuracy score = 0.992
```

Yukarıda ki kod çıktısı incelendiğinde 200 farklı kombinasyon için 5-katlı cross-validation uygulandığında toplamda 1000 tane model fit etme işleminin gerçekleştiğini ve bu işlemlerin 6.5 dakika sürdüğünü görebiliyoruz.

Burada da accuracy skorumuzu 0.992 olarak gözlemledik. Yani daha kısa sürede GridSearchCV ile elde ettiğimiz skoru elde ettik. Bu her zaman eşit çıkmasa da en iyi skora yakın değerler elde etmeyi bekleriz. Dolayısıyla zaman açısından

düşünüldüğünde büyük veri setlerinde RandomizedCV tercih edilebilir.  
İki yöntemin daha detaylı karşılaştırılması için [buraya](#) göz atabilirsiniz.

Ayrıca kodların tamamına [buradan](#) erişebilirsiniz.

Gelecek yazılarda görüşmek üzere...

[Crossvalidation](#)[Python](#)[Sklearn](#)[Hyperparameter Tuning](#)[Makine Öğrenmesi](#)[Follow](#)

## Written by Merve Bayram Durna

1.3K Followers · Writer for Bilişim Hareketi

Data Analyst | Data Scientist

More from Merve Bayram Durna and Bilişim Hareketi



42



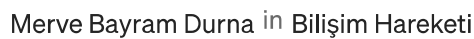
1



13/20

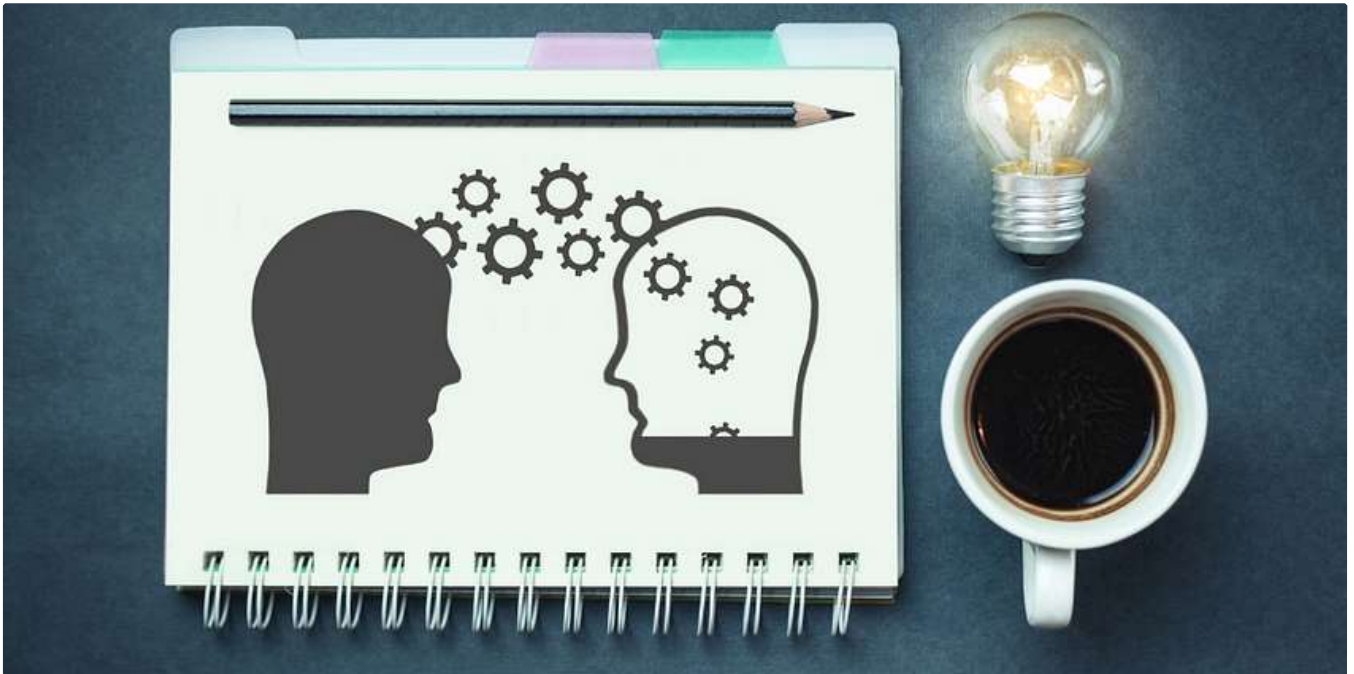


7 min read · Sep 17, 2021



NumPy (Numerical Python) bilimsel hesaplamaları hızlı bir şekilde yapmamızı sağlayan bir matematik kütüphanesidir. Numpy'ın temelini numpy...

...



Merve Bayram Durna

## BERT and Transfer Learning in NLP

The Complete NLP Guide: Text to Context #9

9 min read · Jan 18



61



See all from Merve Bayram Durna

See all from Bilişim Hareketi

## Recommended from Medium



 James Presbitero Jr. in Practice in Public

## These Words Make it Obvious That Your Text is Written By AI

These 7 words are painfully obvious. They make me cringe. They will make your reader cringe.

4 min read · Jan 1

 35K

 908







 Beck Novaes

## The Curse of Knowledge: The Danger of Studying Too Much



## How to Prevent Studies from Turning You into a Fool

19 min read · Jan 27



3.5K



58



### Lists



#### Coding & Development

11 stories · 435 saves



#### Predictive Modeling w/ Python

20 stories · 878 saves



#### Practical Guides to Machine Learning

10 stories · 1020 saves



#### ChatGPT

21 stories · 450 saves



Riikka Iivanainen

## The secret life of people with high self-control (it's easier than you think)

Research suggests that people with high self-control are good at avoiding temptation—not resisting it

8 min read · Jan 9

 8.9K

 152





Kevin Nokia

# Reading Books Is Useless: Here’s a Better Way to Read

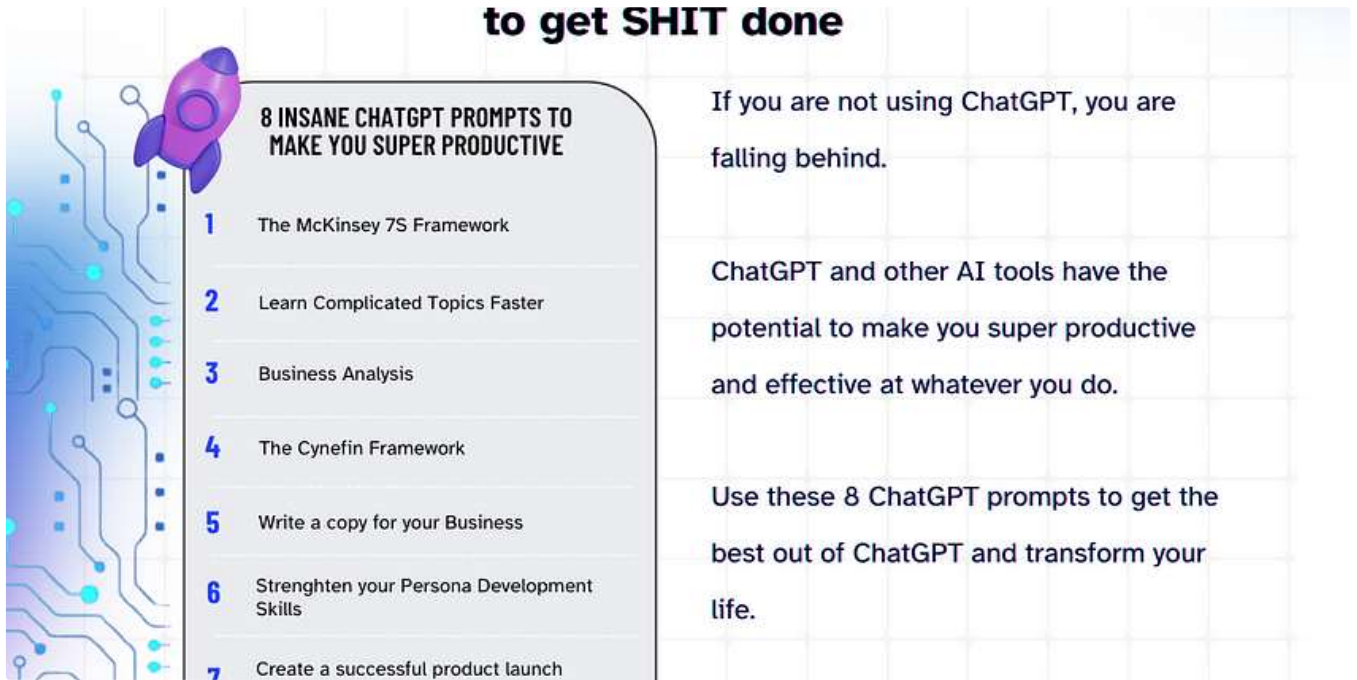
How to become better at reading.

5 min read · Jan 28

 2.7K

 77





Anish Singh Walia in [AI monks.io](#)

## Top 8 ChatGPT Prompts That Will Make You More Productive Than a Team of 20 Employees

AI isn't just artificial; it's authentically driving a productivity revolution. With ChatGPT, efficiency becomes second nature, and...

7 min read · Jan 18



2.9K



37





Thu Vu in Towards Data Science

## How to Learn AI on Your Own (a self-study guide)

If your hands touch a keyboard for work, Artificial Intelligence is going to change your job in the next few years.

★ • 12 min read • Jan 5



3.1K



27



See more recommendations