# DSC 40A: Theoretical Foundations of Data Science

Lecture 13 Part II
**Feature engineering and data transformations**

October 27, 2025

# Announcements

- Gal is out today so Sawyer is lecturing in her place (hence slightly different slides)

# Announcements

- Gal is out today so Sawyer is lecturing in her place (hence slightly different slides)
- Your **midterm exam** will take place Monday, Nov. 3rd!

# Announcements

- Gal is out today so Sawyer is lecturing in her place (hence slightly different slides)
- Your **midterm exam** will take place Monday, Nov. 3rd!
- 50 minutes on paper, no calculators or electronics permitted

# Announcements

- Gal is out today so Sawyer is lecturing in her place (hence slightly different slides)
- Your **midterm exam** will take place Monday, Nov. 3rd!
- 50 minutes on paper, no calculators or electronics permitted
- You are allowed to bring a single double-sided page of notes

# Announcements

- Gal is out today so Sawyer is lecturing in her place (hence slightly different slides)
- Your **midterm exam** will take place Monday, Nov. 3rd!
- 50 minutes on paper, no calculators or electronics permitted
- You are allowed to bring a single double-sided page of notes
- Seats are **assigned**; will provide details during Discussion today and Campuswire afterwards
- Content: Lectures 1-13, Homeworks 1-4, Groupworks 1-5

## Announcements

- Gal is out today so Sawyer is lecturing in her place (hence slightly different slides)
- Your **midterm exam** will take place Monday, Nov. 3rd!
- 50 minutes on paper, no calculators or electronics permitted
- You are allowed to bring a single double-sided page of notes
- Seats are **assigned**; will provide details during Discussion today and Campuswire afterwards
- Content: Lectures 1-13, Homeworks 1-4, Groupworks 1-5
- Prepare by practicing old exam problems on practice.dsc40a.com

# Announcements

Varun and Owen will be hosting a midterm review session this Thursday 10/30 from 5pm-7pm in **Ledden Auditorium** (near HSS/APM)

Stay tuned for further details via Campuswire

# Recap from last week

On **Friday** you covered a few topics that build on our work with simple linear models and multiple regression:

# Recap from last week

On **Friday** you covered a few topics that build on our work with simple linear models and multiple regression:

- Standardizing features $x_{i\ (\text{su})} = \frac{x_i - \overline{x}}{\sigma_x}$,

$$H(x) = w_0 + w_1 x_{1\ (\text{su})} + \ldots + w_d x_{d\ (\text{su})}$$

# Recap from last week

On **Friday** you covered a few topics that build on our work with simple linear models and multiple regression:

- Standardizing features $x_{i\ (\text{su})} = \frac{x_i - \overline{x}}{\sigma_x}$,

$$H(x) = w_0 + w_1 x_{1\ (\text{su})} + \ldots + w_d x_{d\ (\text{su})}$$

- Adding polynomial terms to the hypothesis function, e.g.,

$$H(x) = w_0 + w_1 x + w_2 x^2,$$

# Recap from last week

On **Friday** you covered a few topics that build on our work with simple linear models and multiple regression:

- Standardizing features $x_{i\ (\text{su})} = \frac{x_i - \overline{x}}{\sigma_x}$,

$$H(x) = w_0 + w_1 x_{1\ (\text{su})} + \ldots + w_d x_{d\ (\text{su})}$$

- Adding polynomial terms to the hypothesis function, e.g.,

$$H(x) = w_0 + w_1 x + w_2 x^2,$$

- Adding terms from combinations of features:

$$H(\text{sqft}, \text{comp}) = \ldots + w_4(\text{sqft} \cdot \text{comp}) + \ldots$$

# Recap from last week

On **Friday** you covered a few topics that build on our work with simple linear models and multiple regression:

- Standardizing features $x_{i\ (\text{su})} = \frac{x_i - \bar{x}}{\sigma_x}$,

$$H(x) = w_0 + w_1 x_{1\ (\text{su})} + \ldots + w_d x_{d\ (\text{su})}$$

- Adding polynomial terms to the hypothesis function, e.g.,

$$H(x) = w_0 + w_1 x + w_2 x^2,$$

- Adding terms from combinations of features:

$$H(\text{sqft}, \text{comp}) = \ldots + w_4(\text{sqft} \cdot \text{comp}) + \ldots$$

**Question:** What does each of these have in common?

# Recap from last week

On **Friday** you covered a few topics that build on our work with simple linear models and multiple regression:

- Standardizing features $x_{i\ (\text{su})} = \frac{x_i - \bar{x}}{\sigma_x}$,

$$H(x) = w_0 + w_1 x_{1\ (\text{su})} + \ldots + w_d x_{d\ (\text{su})}$$

- Adding polynomial terms to the hypothesis function, e.g.,

$$H(x) = w_0 + w_1 x + w_2 x^2,$$

- Adding terms from combinations of features:

$$H(\text{sqft}, \text{comp}) = \ldots + w_4(\text{sqft} \cdot \text{comp}) + \ldots$$

**Question:** What does each of these have in common?

# Recap from last week

On **Friday** you covered a few topics that build on our work with simple linear models and multiple regression:

- Standardizing features $x_{i\ (\text{su})} = \frac{x_i - \overline{x}}{\sigma_x}$,

$$H(x) = w_0 + w_1 x_{1\ (\text{su})} + \ldots + w_d x_{d\ (\text{su})}$$

- Adding polynomial terms to the hypothesis function, e.g.,

$$H(x) = w_0 + w_1 x + w_2 x^2,$$

- Adding terms from combinations of features:

$$H(\text{sqft}, \text{comp}) = ... + w_4(\text{sqft} \cdot \text{comp}) + ...$$

**Question:** What do each of these have in common?

These are **all linear** in the weights $w_i$.

What if we want to use a hypothesis function that is **nonlinear in the weights and/or features?**

What if we want to use a hypothesis function that is **nonlinear in the weights and/or features?**

**Example** A nonlinear hypothesis function

Consider the following hypothesis function, which depends on a single scalar-valued feature and two weights $w_0, w_1$:

$$H(x) = w_0 e^{w_1 x}.$$

What if we want to use a hypothesis function that is **nonlinear in the weights and/or features?**

**Example** A nonlinear hypothesis function

Consider the following hypothesis function, which depends on a single scalar-valued feature and two weights $w_0, w_1$:

$$H(x) = w_0 e^{w_1 x}.$$

This function is **nonlinear** in both the weights and the feature $x$. We can create a new hypothesis function $T(x) = b_0 + b_1 x$, which is linear in the weights $b_0, b_1$, by applying the transformation

$$T(x) = \ln(H(x)) = \ln(w_0) + w_1 x.$$

What if we want to use a hypothesis function that is **nonlinear in the weights and/or features?**

**Example** A nonlinear hypothesis function

Consider the following hypothesis function, which depends on a single scalar-valued feature and two weights $w_0, w_1$:

$$H(x) = w_0 e^{w_1 x}.$$

This function is **nonlinear** in both the weights and the feature $x$. We can create a new hypothesis function $T(x) = b_0 + b_1 x$, which is linear in the weights $b_0, b_1$, by applying the transformation

$$T(x) = \ln(H(x)) = \ln(w_0) + w_1 x.$$

The weights are related by the equations $b_0 = \ln(w_0)$ and $b_1 = w_1$.

**Example** A nonlinear hypothesis function

$$T(x) = \ln(H(x)) = \ln(\mathbf{w_0}) + \mathbf{w_1} x$$

Then, we can fit the linear hypothesis function $T(x) = b_0 + b_1 x$ to data using the normal equations to obtain optimal weights $b_0^*, b_1^*$.

**Example** A nonlinear hypothesis function

$$T(x) = \ln(H(x)) = \ln(\mathbf{w_0}) + \mathbf{w_1} x$$

Then, we can fit the linear hypothesis function $T(x) = b_0 + b_1 x$ to data using the normal equations to obtain optimal weights $b_0^*, b_1^*$.Finally, we can recover the optimal weights for the original hypothesis via

$$w_0^* = e^{b_0^*},$$
$$w_1^* = b_1^*.$$

**Example** A nonlinear hypothesis function

$$T(x) = \ln(H(x)) = \ln(\mathbf{w_0}) + \mathbf{w_1} x$$

Then, we can fit the linear hypothesis function $T(x) = b_0 + b_1 x$ to data using the normal equations to obtain optimal weights $b_0^*, b_1^*$.Finally, we can recover the optimal weights for the original hypothesis via

$$w_0^* = e^{b_0^*},$$
$$w_1^* = b_1^*.$$

We will explore this in an interactive notebook, continuing from last week's example.

Let's do a more detailed example.

Let's do a more detailed example.

**Example** Drink up!

You operate a beverage bottling plant in the Southwestern US. Recently you collected data over the course of twelve weeks $i = 1, \ldots, 12$, capturing the following statistics:

Let's do a more detailed example.

**Example** Drink up!

You operate a beverage bottling plant in the Southwestern US. Recently you collected data over the course of twelve weeks $i = 1, \ldots, 12$, capturing the following statistics:

- $x_i^{(1)}$, the labor-hours of the plant during week $i$,

Let's do a more detailed example.

**Example** Drink up!

You operate a beverage bottling plant in the Southwestern US. Recently you collected data over the course of twelve weeks $i = 1, \ldots, 12$, capturing the following statistics:

- $x_i^{(1)}$, the labor-hours of the plant during week $i$,
- $x_i^{(2)}$, the electricity consumed in the plant during week $i$,

Let's do a more detailed example.

**Example** Drink up!

You operate a beverage bottling plant in the Southwestern US. Recently you collected data over the course of twelve weeks $i = 1, \ldots, 12$, capturing the following statistics:

- $x_i^{(1)}$, the labor-hours of the plant during week $i$,
- $x_i^{(2)}$, the electricity consumed in the plant during week $i$,
- $x_i^{(3)}$, the materials input (kg of syrup/concentrate), again during week $i$.

Let's do a more detailed example.

**Example** Drink up!

You operate a beverage bottling plant in the Southwestern US. Recently you collected data over the course of twelve weeks $i = 1, \ldots, 12$, capturing the following statistics:

- $x_i^{(1)}$, the labor-hours of the plant during week $i$,
- $x_i^{(2)}$, the electricity consumed in the plant during week $i$,
- $x_i^{(3)}$, the materials input (kg of syrup/concentrate), again during week $i$.

You would like to model $y_i$, the liters of finished bottled product during week $i$, in terms of measurable quantities.

**Example** Drink up!

After discussing things with your in-house econometrics guru, you arrive at the following hypothesis function:

$$H(\vec{x}) = w_0 (x^{(1)})^{w_1} (x^{(2)})^{w_2} (x^{(3)})^{w_3}.$$

**Example** Drink up!

After discussing things with your in-house econometrics guru, you arrive at the following hypothesis function:

$$H(\vec{x}) = w_0 (x^{(1)})^{w_1} (x^{(2)})^{w_2} (x^{(3)})^{w_3}.$$

This is an example of a **Cobb-Douglas** production function, commonly used in economics to model output as a function of multiple inputs.

**Example** Drink up!

After discussing things with your in-house econometrics guru, you arrive at the following hypothesis function:

$$H(\vec{x}) = w_0 (x^{(1)})^{w_1} (x^{(2)})^{w_2} (x^{(3)})^{w_3}.$$

This is an example of a **Cobb-Douglas** production function, commonly used in economics to model output as a function of multiple inputs.

Note that this hypothesis function is **nonlinear** in both the weights $w_i$ and the features $x^{(j)}$.

**Example** Drink up!

After discussing things with your in-house econometrics guru, you arrive at the following hypothesis function:

$$H(\vec{x}) = w_0 (x^{(1)})^{w_1} (x^{(2)})^{w_2} (x^{(3)})^{w_3}.$$

This is an example of a **Cobb-Douglas** production function, commonly used in economics to model output as a function of multiple inputs.

Note that this hypothesis function is **nonlinear** in both the weights $w_i$ and the features $x^{(j)}$.

To fit this model to data, we will need to perform a transformation. By using a logarithm, we can obtain a new hypothesis function $T(\vec{x})$ that is linear in the weights:

**Example** Drink up!

After discussing things with your in-house econometrics guru, you arrive at the following hypothesis function:

$$H(\vec{x}) = w_0 (x^{(1)})^{w_1} (x^{(2)})^{w_2} (x^{(3)})^{w_3}.$$

This is an example of a **Cobb-Douglas** production function, commonly used in economics to model output as a function of multiple inputs.

Note that this hypothesis function is **nonlinear** in both the weights $w_i$ and the features $x^{(j)}$.

To fit this model to data, we will need to perform a transformation. By using a logarithm, we can obtain a new hypothesis function $T(\vec{x})$ that is linear in the weights:

$$\begin{aligned} T(\vec{x}) &= \ln(H(\vec{x})) \\ &= \ln(w_0) + w_1 \ln(x^{(1)}) + w_2 \ln(x^{(2)}) + w_3 \ln(x^{(3)}). \end{aligned}$$

**Example** Drink up!

$$\begin{aligned} T(\vec{x}) &= \ln(H(\vec{x})) \\ &= \ln(\mathbf{w_0}) + \mathbf{w_1} \ln(x^{(1)}) + \mathbf{w_2} \ln(x^{(2)}) + \mathbf{w_3} \ln(x^{(3)}). \end{aligned}$$

We can now fit the linear hypothesis function

$$T(\vec{x}) = b_0 + b_1 z^{(1)} + b_2 z^{(2)} + b_3 z^{(3)},$$

**Example** Drink up!

$$
\begin{aligned}
T(\vec{x}) &= \ln(H(\vec{x})) \\
&= \ln(\mathbf{w_0}) + \mathbf{w_1} \ln(x^{(1)}) + \mathbf{w_2} \ln(x^{(2)}) + \mathbf{w_3} \ln(x^{(3)}).
\end{aligned}
$$

We can now fit the linear hypothesis function

$$T(\vec{x}) = b_0 + b_1 z^{(1)} + b_2 z^{(2)} + b_3 z^{(3)},$$

where we have defined the transformed features

$$z^{(j)} = \ln(x^{(j)}), \quad j = 1, 2, 3,$$

using the normal equations to obtain optimal weights $b_0^*, b_1^*, b_2^*, b_3^*$.

**Example** Drink up!

$$T(\vec{x}) = \ln(H(\vec{x}))$$
$$= \ln(\mathbf{w_0}) + \mathbf{w_1} \ln(x^{(1)}) + \mathbf{w_2} \ln(x^{(2)}) + \mathbf{w_3} \ln(x^{(3)}).$$

We can now fit the linear hypothesis function

$$T(\vec{x}) = b_0 + b_1 z^{(1)} + b_2 z^{(2)} + b_3 z^{(3)},$$

where we have defined the transformed features

$$z^{(j)} = \ln(x^{(j)}), \quad j = 1, 2, 3,$$

using the normal equations to obtain optimal weights $b_0^*, b_1^*, b_2^*, b_3^*$.

**Example** Drink up!

Finally, we can recover the optimal weights for the original hypothesis via

$$
\begin{aligned}
w_0^* &= e^{b_0^*}, \\
w_1^* &= b_1^*, \\
w_2^* &= b_2^*, \\
w_3^* &= b_3^*.
\end{aligned}
$$

Note: unlike before, we need to transform our *features* as well as our *weights*!

# Question

Answer at **q.dsc40a.com**.

# Question

Answer at **q.dsc40a.com**.

Which of the following hypothesis functions is **not** linear in the parameters?

(A) $H(\vec{x}) = w_1 \left(x^{(1)} x^{(2)}\right) + \frac{w_2}{x^{(1)}} \sin\left(x^{(2)}\right)$

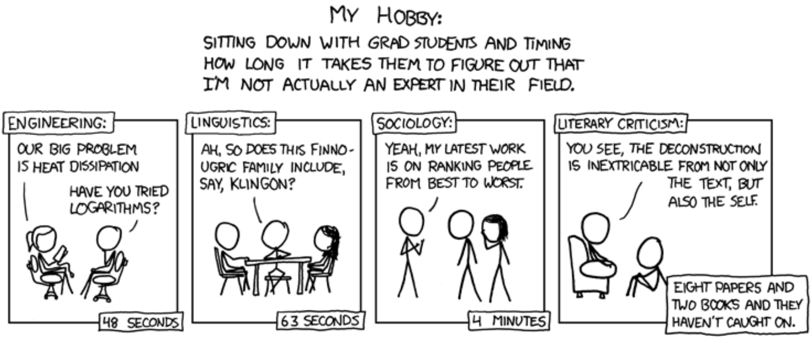
(B) $H(\vec{x}) = 2^{\,w_1}\, x^{(1)}$

(C) $H(\vec{x}) = \vec{w} \cdot \text{Aug}(\vec{x})$

(D) $H(\vec{x}) = w_1 \cos\left(x^{(1)}\right) + w_2\, 2^{\,x^{(2)} \log x^{(3)}}$

(E) More than one of the above.

# Have you tried using logarithms?




**xkcd #451**

Sometimes, it's just not possible to transform a hypothesis function to be linear in terms of some parameters.

Sometimes, it's just not possible to transform a hypothesis function to be linear in terms of some parameters.

In those cases, you'd have to resort to other methods of finding the optimal parameters.

Sometimes, it's just not possible to transform a hypothesis function to be linear in terms of some parameters.

In those cases, you'd have to resort to other methods of finding the optimal parameters.

- For example, $H(x) = w_0 \sin(w_1 x)$ *can't* be transformed to be linear.

Sometimes, it's just not possible to transform a hypothesis function to be linear in terms of some parameters.

In those cases, you'd have to resort to other methods of finding the optimal parameters.

- For example, $H(x) = w_0 \sin(w_1 x)$ *can't* be transformed to be linear.
- But there are other methods of minimizing mean squared error:

Sometimes, it's just not possible to transform a hypothesis function to be linear in terms of some parameters.

In those cases, you'd have to resort to other methods of finding the optimal parameters.

- For example, $H(x) = w_0 \sin(w_1 x)$ *can't* be transformed to be linear.
- But there are other methods of minimizing mean squared error:

$$R_{\text{sq}}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} \big(y_i - w_0 \sin(w_1 x)\big)^2.$$

- One method: **gradient descent**, the topic of the next lecture!

Sometimes, it's just not possible to transform a hypothesis function to be linear in terms of some parameters.

In those cases, you'd have to resort to other methods of finding the optimal parameters.

- For example, $H(x) = w_0 \sin(w_1 x)$ *can't* be transformed to be linear.
- But there are other methods of minimizing mean squared error:

$$R_{\text{sq}}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - w_0 \sin(w_1 x)\right)^2.$$

- One method: **gradient descent**, the topic of the next lecture!

Hypothesis functions that are linear in the parameters are much easier to work with.

# Roadmap

- This is the end of the content that's in scope for the Midterm Exam.

# Roadmap

- This is the end of the content that's in scope for the Midterm Exam.
- Now, we'll introduce **gradient descent**, a technique for minimizing functions that can't be minimized directly using calculus or linear algebra.

# Roadmap

- This is the end of the content that's in scope for the Midterm Exam.
- Now, we'll introduce **gradient descent**, a technique for minimizing functions that can't be minimized directly using calculus or linear algebra.
- After the Midterm Exam, we'll switch gears to **probability theory**.

# DSC 40A: Theoretical Foundations of Data Science

Lecture 14 Part I
**Gradient Descent**

October 27, 2025

# Minimizing empirical risk

- Repeatedly, we've been tasked with **minimizing** the value of empirical risk functions.

# Minimizing empirical risk

- Repeatedly, we've been tasked with **minimizing** the value of empirical risk functions.
  - Why? To help us find the **best** model parameters, $h^*$ or $\vec{w}^*$, which help us make the **best** predictions!

# Minimizing empirical risk

- Repeatedly, we've been tasked with **minimizing** the value of empirical risk functions.
    - Why? To help us find the **best** model parameters, $h^*$ or $\vec{w}^*$, which help us make the **best** predictions!
- We've minimized empirical risk functions in various ways.

# Minimizing empirical risk

- Repeatedly, we've been tasked with **minimizing** the value of empirical risk functions.
  - Why? To help us find the **best** model parameters, $h^*$ or $\vec{w}^*$, which help us make the **best** predictions!
- We've minimized empirical risk functions in various ways.
  - $R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h)^2$

# Minimizing empirical risk

- Repeatedly, we've been tasked with **minimizing** the value of empirical risk functions.
  - Why? To help us find the **best** model parameters, $h^*$ or $\vec{w}^*$, which help us make the **best** predictions!
- We've minimized empirical risk functions in various ways.
  - $R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h)^2$ **critical points where $R' = 0$**
  - $R_{\text{abs}}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} |\, y_i - (w_0 + w_1 x)|$

# Minimizing empirical risk

- Repeatedly, we've been tasked with **minimizing** the value of empirical risk functions.
  - Why? To help us find the **best** model parameters, $h^*$ or $\vec{w}^*$, which help us make the **best** predictions!
- We've minimized empirical risk functions in various ways.
  - $R_{\text{sq}}(h) = \frac{1}{n}\sum_{i=1}^{n}(y_i - h)^2$ **critical points where $R' = 0$**
  - $R_{\text{abs}}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n} \left| y_i - (w_0 + w_1 x)\right|$ **Brute force (Hw3, P7)**
  - $R_{\text{sq}}(\vec{w}) = \frac{1}{n}\|\vec{y} - X\vec{w}\|^2$

# Minimizing empirical risk

- Repeatedly, we've been tasked with **minimizing** the value of empirical risk functions.
  - Why? To help us find the **best** model parameters, $h^*$ or $\vec{w}^*$, which help us make the **best** predictions!
- We've minimized empirical risk functions in various ways.
  - $R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h)^2$ **critical points where $R' = 0$**
  - $R_{\text{abs}}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} \left| y_i - (w_0 + w_1 x) \right|$ **Brute force (Hw3, P7)**
  - $R_{\text{sq}}(\vec{w}) = \frac{1}{n} \|\vec{y} - X\vec{w}\|^2$ **projections or $\nabla R = \vec{0}$**

# Minimizing arbitrary functions

- Assume $f(t)$ is some *differentiable* single-variable function.

# Minimizing arbitrary functions

- Assume $f(t)$ is some *differentiable* single-variable function.
- When tasked with minimizing $f(t)$, our general strategy has been to:

# Minimizing arbitrary functions

- Assume $f(t)$ is some *differentiable* single-variable function.
- When tasked with minimizing $f(t)$, our general strategy has been to:
  1. Find $\frac{df}{dt}(t)$, the derivative of $f$.

# Minimizing arbitrary functions

- Assume $f(t)$ is some *differentiable* single-variable function.
- When tasked with minimizing $f(t)$, our general strategy has been to:
  1. Find $\frac{df}{dt}(t)$, the derivative of $f$.
  2. Find the input $t^*$ such that $\frac{df}{dt}(t^*) = 0$.

# Minimizing arbitrary functions

- Assume $f(t)$ is some *differentiable* single-variable function.
- When tasked with minimizing $f(t)$, our general strategy has been to:
  1. Find $\frac{df}{dt}(t)$, the derivative of $f$.
  2. Find the input $t^*$ such that $\frac{df}{dt}(t^*) = 0$.
  3. Check that $\frac{d^2f}{dt^2}(t^*) > 0$ so that $t^*$ is a true minimizer.

# Minimizing arbitrary functions

- Assume $f(t)$ is some *differentiable* single-variable function.
- When tasked with minimizing $f(t)$, our general strategy has been to:
  1. Find $\frac{df}{dt}(t)$, the derivative of $f$.
  2. Find the input $t^*$ such that $\frac{df}{dt}(t^*) = 0$.
  3. Check that $\frac{d^2f}{dt^2}(t^*) > 0$ so that $t^*$ is a true minimizer.
- However, there are cases where we can find $\frac{df}{dt}(t)$, but it is **either difficult or impossible** to solve $\frac{df}{dt}(t^*) = 0$.

# Minimizing arbitrary functions

- Assume $f(t)$ is some *differentiable* single-variable function.
- When tasked with minimizing $f(t)$, our general strategy has been to:
  1. Find $\frac{df}{dt}(t)$, the derivative of $f$.
  2. Find the input $t^*$ such that $\frac{df}{dt}(t^*) = 0$.
  3. Check that $\frac{d^2f}{dt^2}(t^*) > 0$ so that $t^*$ is a true minimizer.
- However, there are cases where we can find $\frac{df}{dt}(t)$, but it is **either difficult or impossible** to solve $\frac{df}{dt}(t^*) = 0$.

$$\frac{df}{dt}(t) = 5t^4 - t^3 - 5t^2 + 2t - 9$$

- Then what?

When we can't directly solve for the minimizer of a function, we can **approximate** the minimizer using an iterative method called **gradient descent**.

When we can't directly solve for the minimizer of a function, we can **approximate** the minimizer using an iterative method called **gradient descent**.

The idea is to start at some initial guess $t_0$ and then **iteratively improve** our guess by taking steps in the direction of steepest descent (i.e., the negative gradient).

When we can't directly solve for the minimizer of a function, we can **approximate** the minimizer using an iterative method called **gradient descent**.

The idea is to start at some initial guess $t_0$ and then **iteratively improve** our guess by taking steps in the direction of steepest descent (i.e., the negative gradient).

Over time, these steps will (hopefully) lead us to a point close to the true minimizer $t^*$.

When we can't directly solve for the minimizer of a function, we can **approximate** the minimizer using an iterative method called **gradient descent**.

The idea is to start at some initial guess $t_0$ and then **iteratively improve** our guess by taking steps in the direction of steepest descent (i.e., the negative gradient).

Over time, these steps will (hopefully) lead us to a point close to the true minimizer $t^*$.