

# Week 1 - Linear and Logistic Regression

```
In [2]: import pandas as pd
import numpy as np
from sklearn import datasets
import seaborn as sns
import matplotlib.pyplot as plt

import math
import scipy.stats as stats
```

## Module 1 - Assignment 1: Linear Regression

### Instructions

Review the follow scenario and attach your responses in a word document:

Scenario: You are part of a smartwatch startup. Your marketing team has been running advertisements across multiple channels (social media, TV, print). The CFO wants to predict revenue for the next month based on ad spend data. Your task is to use linear regression to predict revenue and analyze the impact of increasing the advertising budget.

Question:

You have the following data:

Ad Spend (*in1000s*) *Revenue*( in 1000s)

10 150

20 290

30 440

40 600

50 720

Use linear regression to find the relationship between ad spend and revenue.

What is the expected revenue if the ad spend is \$60,000?

What does the slope of the line represent in this context?

```
In [6]: import statsmodels.api as sm

add = np.array([10,20,30,40,50])
revenue = np.array([150,290,440,600,720])

model = sm.OLS(revenue, sm.add_constant(add))
```

```
results = model.fit()

residuals = results.resid
```

```
In [11]: results.params
```

```
Out[11]: array([ 5. , 14.5])
```

```
In [13]: add_=60
rev = (add_*results.params[1])+results.params[0]
print(rev)
```

```
875.0000000000002
```

Using linear regression we get the following model:

$$revenue_i = add_i * 14.5 + 5 + error_i$$

- From this model we can infer that if we spend 60,000 we would get revenue of 875,000
- The slope is the rate of how much dollars would the revenue increase if we increase in 1000 dollars in add spends. So if you spend 1000 dollars, the revenue would be \$19,500.

## Module 1 - Assignment 2: Logistic Regression

### Instructions

Review the following scenario and attach your responses in a word document:

Scenario: You are managing a fitness subscription platform. The company wants to predict whether users will renew their subscriptions based on their activity levels. The dataset contains user activity data and renewal status.

Question:

You have the following data:

Weekly Usage Hours    Renewal Status (1 = Yes, 0 = No)

1 0

3 0

5 1

7 1

9 1

Use logistic regression to model the probability of renewal based on weekly usage hours.

What is the probability that a user logging 4 hours per week will renew their subscription?

Based on the model, recommend a threshold for targeting at-risk users with promotions.

```
In [17]: import statsmodels.api as sm

act = np.array([1,3,5,7,9])
renewal = np.array([0,0,1,1,1])

model = sm.Logit(renewal, sm.add_constant(act))

results = model.fit()
```

Warning: Maximum number of iterations has been exceeded.  
Current function value: 0.000000  
Iterations: 35

/Users/brunocavagnaro/MS\_PURDUE\_DS/GRAD\_50900\_AppliedMachineLearning/app\_m  
l/lib/python3.10/site-packages/statsmodels/discrete/discrete\_model.py:227:  
PerfectSeparationWarning: Perfect separation or prediction detected, param  
eter may not be identified  
warnings.warn(msg, category=PerfectSeparationWarning)  
/Users/brunocavagnaro/MS\_PURDUE\_DS/GRAD\_50900\_AppliedMachineLearning/app\_m  
l/lib/python3.10/site-packages/statsmodels/base/model.py:607: ConvergenceW  
arning: Maximum Likelihood optimization failed to converge. Check mle\_retv  
als  
warnings.warn("Maximum Likelihood optimization failed to "

```
In [20]: results.params
```

```
Out[20]: array([-80.38287523,  21.27382992])
```

```
In [39]: act_ = 4
linear = (act_*results.params[1]) + results.params[0]
prob = 1 / (1 + math.exp(-linear))
```

```
In [38]: prob
```

```
Out[38]: 0.9999999999948204
```

Using logistic regression we get the following model:

$$linear_i = activity_i * 21.274 - 80.383$$

$$Renewal_i = \frac{1}{1 + e^{-linear_i}}$$

- From this model we can infer that if we spend 60,000 we would get revenue of 875,000
- The slope is the rate of how much dollars would the revenue increase if we increase in 1000 dollars in add spends. So if you spend 1000 dollars, the revenue would be \$19,500.

# Module 1 - Assignment 3: Understanding the difference between Linear and Logistic Regression

## Instructions

Review the follow scenario and attach your responses in a word document:

Scenario: You are part of a data science team at a health-tech startup. The company is launching two initiatives:

Predicting Patient Recovery Time: Based on historical data, the company wants to predict the number of days it takes for a patient to recover from surgery.

Predicting Disease Diagnosis: The company also wants to predict whether a patient has a specific disease based on diagnostic test results (positive/negative).

Your task is to decide which regression model—linear or logistic regression—is appropriate for each initiative, justify the choice, and explain the differences between these methods.

Question:

Which regression model (linear or logistic) is suitable for each of the two initiatives? Justify your choice.

Explain the key differences between linear and logistic regression, focusing on:

The type of dependent variable they model.

The mathematical equations used.

Real-world scenarios where each is used.

Assume you are building a model for Disease Diagnosis. Based on the logistic regression equation:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Calculate the probability of a disease diagnosis for a patient with a test score  $x = 5$ , given  $\beta_0 = -2$  and  $\beta_1 = 0.8$ .

In [43]: `1/(1+math.exp(-(-2+(0.8*5))))`

Out [43]: `0.8807970779778823`

## Answer

**Which regression model (linear or logistic) is suitable for each of the two initiatives? Justify your choice.**

- **Predicting Patient Recovery Time:** As predicting the recovery time of a patient is a continuous dependent variable, the most appropriate model is the linear regression.
- **Predicting Disease Diagnosis:** As the disease diagnosis of a patient is a binary dependent variable, the most appropriate is the logistic regression model.

**Explain the key differences between linear and logistic regression, focusing on:**

- Linear regression: Models a continuous outcome (any real number). Examples: price, temperature, revenue, blood pressure.

$$y = \beta_0 + \beta_1 x$$

- Logistic regression: Models a categorical outcome—most commonly binary (0/1). Examples: churn vs. not churn, disease vs. no disease, click vs. no click, default vs. not default

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

**Assume you are building a model for Disease Diagnosis. Based on the logistic regression equation:**

Calculate the probability of a disease diagnosis for a patient with a test score  $x = 5$ , given  $\beta_0 = -2$  and  $\beta_1 = 0.8$ .

The probability is equal to **88%**

## Module 1 - Assignment 4: Review the Reading

### Instructions

Review and Read the Paper "Learning Multiple Layers of Representation" in the reading section of this module and submit your answers to the questions and submit in a word document.

About Geoffrey Hinton

Geoffrey Hinton is a trailblazer in artificial intelligence and a key figure in the development of deep learning. A British-Canadian cognitive psychologist and computer scientist, Hinton has made transformative contributions to neural networks, shaping how machines learn to recognize patterns and make intelligent decisions. His work is foundational to modern AI applications such as computer vision, natural language processing, and autonomous systems.

Hinton's most notable achievement is his pioneering work on the backpropagation algorithm, which made it computationally feasible to train deep neural networks by iteratively adjusting weights to minimize error. This algorithm, though mathematically established earlier, gained practical relevance due to Hinton's insights into how it could be applied effectively in multi-layered networks. Backpropagation underpins

nearly all modern neural network architectures, from convolutional networks (used in image recognition) to transformers (used in language models like ChatGPT).

In 2018, Hinton was honored with the Turing Award, the “Nobel Prize of Computing,” alongside Yoshua Bengio and Yann LeCun. The award recognized their breakthroughs in deep learning, which provided the foundation for AI systems that now excel in tasks such as speech recognition, language translation, medical diagnostics, and protein folding (as demonstrated by AlphaFold). This trio, often called the “Godfathers of Deep Learning,” helped push deep neural networks from theoretical constructs to practical tools capable of transforming industries. Hinton’s work also extends beyond backpropagation. He introduced concepts like restricted Boltzmann machines (RBMs) and deep belief networks (DBNs), which were some of the earliest generative models that could efficiently learn hierarchical representations of data. His research emphasized the importance of unsupervised and generative learning, which allows AI systems to learn from unlabelled data, a critical step toward making machine learning more scalable and adaptable. Hinton’s influence extends into both academia and industry. He co-founded the Google Brain team and continues to mentor researchers, fostering innovation at the intersection of AI and cognitive science.

## Summary

This paper, “Learning multiple layers of representation” by Geoffrey Hinton, discusses the foundational concept of deep learning—specifically, how neural networks can learn hierarchical feature representations by leveraging multiple layers. It introduces the concept of generative models, which aim to learn how to generate data by modeling the underlying structure, as opposed to just classifying it (as in purely discriminative models like traditional supervised approaches).

## Key Concepts in the Paper

### Feature Learning in Neural Networks

Traditional backpropagation required large amounts of labeled data and was difficult to apply effectively in deep networks. Hinton describes how neural networks with “top-down” generative connections can use unlabeled data effectively by reconstructing sensory input, enabling efficient feature learning. Generative models, like the ones illustrated in the paper, infer hidden representations (latent variables) and reconstruct data, modeling the joint distribution  $P(x,y)$  of input and output. In contrast, discriminative models, such as linear or logistic regression, directly predict  $P(y|x)$ , focusing solely on classification tasks. Networks with both top-down and bottom-up connections leverage feedback to generate data and refine their understanding of the sensory world, mimicking the hierarchical processing of the human visual system. The paper demonstrates the power of generative models in handwritten digit classification, where the network learns to infer representations from input images, even under conditions with limited labeled data.

## Two Questions Based on the Paper

### Theoretical Question:

In the context of this paper, how do generative models overcome the limitations of traditional supervised learning (like logistic regression) in scenarios with limited labeled data? Provide an explanation considering the joint distribution  $P(x,y)$  versus conditional probability  $P(y|x)$ .

Hint: Consider how generative models leverage unlabeled data to learn feature representations and compare that to the reliance on labeled data in logistic regression.

### Applied Question:

Refer to the figure in the paper illustrating a neural network with multiple hidden layers (e.g., 2000, 500, 500 units). If you were to modify this network to perform a logistic regression task for binary classification (e.g., predicting whether a digit is odd or even), what changes would you make to the network architecture? Why?

Hint: Think about how the final layer in a network is adapted for binary classification tasks and how it relates to the sigmoid activation used in logistic regression.

## Answer:

### Question 1 — Theoretical

*In the context of this paper, how do generative models overcome the limitations of traditional supervised learning (like logistic regression) in scenarios with limited labeled data? Provide an explanation considering the joint distribution  $P(x, y)$  versus the conditional probability  $P(y | x)$ .*

Traditional supervised learners such as logistic regression focus directly on the conditional probability  $P(y | x)$ . This requires sufficient labeled data to learn a good decision boundary. When labels are scarce,  $P(y | x)$  is weakly constrained, leading to high variance and poor generalization.

Generative models, by contrast, aim to explain how the data are produced. They model the joint distribution  $P(x, y)$  (often factorized as  $P(x | y)P(y)$ ). By learning to reconstruct or generate inputs, these models extract structure from large quantities of unlabeled data: regularities, latent factors, and hierarchical features. Once this representation is learned, only a small number of labels are needed to align features with classes; Bayes' rule can recover  $P(y | x)$  from the learned generative components. In Hinton's formulation, greedy layer-wise generative pretraining learns each layer to model its inputs, producing distributed representations that make subsequent supervised fine-tuning easier, more stable, and less label-hungry.

### Question 2 — Applied

*Refer to the figure in the paper illustrating a neural network with multiple hidden layers (e.g., 2000, 500, 500 units). If you were to modify this network to perform a*

*logistic regression task for binary classification (e.g., predicting whether a digit is odd or even), what changes would you make to the network architecture? Why?*

Keep the multi-layer feature extractor but replace the generative top with a discriminative head:

1. Use the pretrained hidden layers (e.g.,  $2000 \rightarrow 500 \rightarrow 500$ ) as a feature encoder. You may freeze them initially or allow fine-tuning.
2. Add a single output unit with a sigmoid activation on top of the highest hidden layer to model  $P(y = 1 | x)$ . This is equivalent to a logistic regression layer operating on learned features.
3. Train this final layer (and optionally fine-tune lower layers) using binary cross-entropy (log-loss).

Generative pretraining discovers useful, low-dimensional representations from unlabeled data. The sigmoid output layer converts the model into a binary classifier consistent with logistic regression, while cross-entropy directly optimizes conditional likelihood  $P(y | x)$ . If desired, the generative decoder/top-down connections used during pretraining can be removed during discriminative fine-tuning, or retained only for auxiliary reconstruction losses.