Day 1: Introduction to Machine Learning

Summer STEM: Machine Learning

Nikola Janjušević, Akshaj Kumar Veldanda, Jacky Yuan, Tejaishwarya Gagadam

Department of Electrical and Computer Engineering
NYU Tandon School of Engineering
Brooklyn, New York

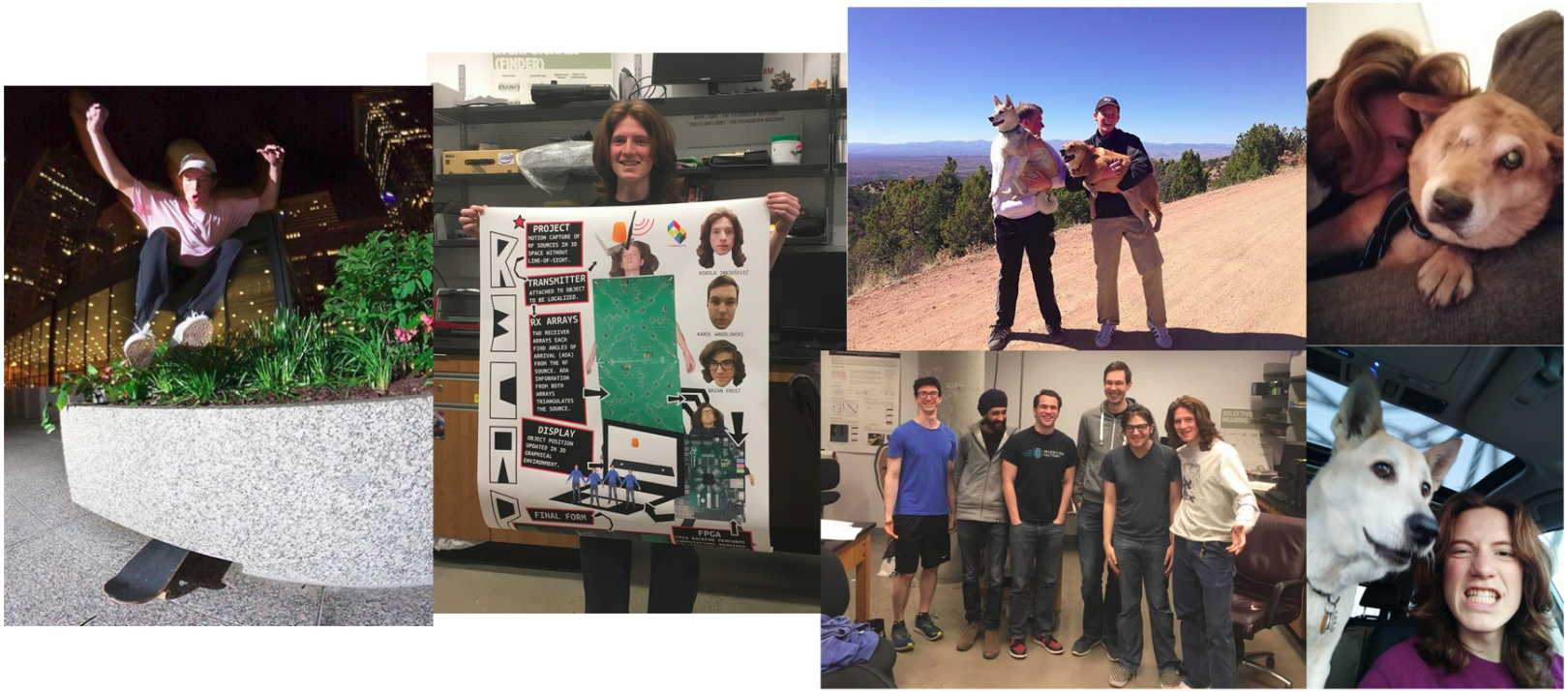
July 8, 2019

NYU TANDON SCHOOL OF ENGINEERING

Outline

NYU TANDON SCHOOL OF ENGINEERING

Nikola



Akshaj

Aishwarya

Jacky

Tell the class about yourself

- Write down the following pieces of information on your notecard:
  - Name
  - Grade
  - Where are you from?
  - What do you want to get out of this class?
  - What time did you wake up this morning (exact times please!)?
  - Where did you travel from?
  - What mode of transport did you take?
  - How long did it take you? (exact times please!)
  - One rule you'd like to propose for the classroom
- Stand in front of the class and present what you've written down
- We'll visualize this data in python later today
  - Link to excel sheet here

Outline

NYU TANDON SCHOOL OF ENGINEERING

Machine Learning

- Most recent exciting technology

Machine Learning

- Most recent exciting technology
- We use these algorithms dozens of times a day

Machine Learning

- Most recent exciting technology
- We use these algorithms dozens of times a day
  - Web search engine

Machine Learning

- Most recent exciting technology
- We use these algorithms dozens of times a day
  - Web search engine
  - Face detection

Machine Learning

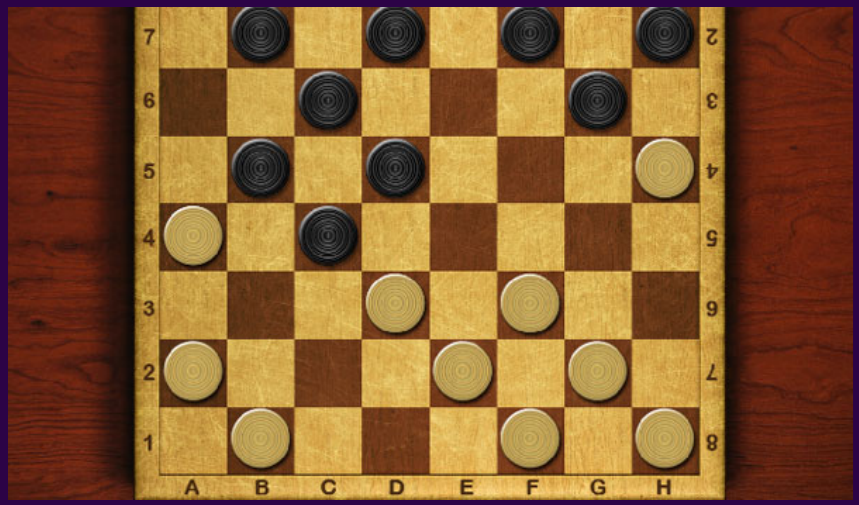
- Most recent exciting technology
- We use these algorithms dozens of times a day
  - Web search engine
  - Face detection
- Machine Learning is an important component to achieve the big AI dream

Machine Learning

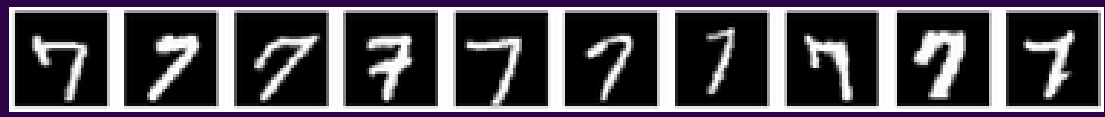
- Most recent exciting technology
- We use these algorithms dozens of times a day
  - Web search engine
  - Face detection
- Machine Learning is an important component to achieve the big AI dream
- Practice is the key to learn machine learning

Definition

- Machine Learning is a field of study that gives computers the ability to learn without being explicitly programmed.



Example: Digit Recognition



- Challenges with expert approach
  - Simple expert rule breaks down in practice
  - Difficult to translate our knowledge into code
- Machine Learning approach
  - Learned systems do very well on image recognition problems

```
def classify(image):
    ...
    nv = count_vert_lines(image)
    nh = count_horiz_lines(image)
    ...

    if (nv == 1) and (nh == 1):
        digit = 7
    ...

    return digit
```

NYU TANDON SCHOOL OF ENGINEERING

Machine Learning Problem Pipeline

1. Gather data

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data
3. Formulate ML problem

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data
3. Formulate ML problem
    - Regression vs Classification

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data
3. Formulate ML problem
   - Regression vs Classification
   - Choose an appropriate cost function

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data
3. Formulate ML problem
   - Regression vs Classification
   - Choose an appropriate cost function
4. Design the model and train to find the optimal parameters of the model

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data
3. Formulate ML problem
   - Regression vs Classification
   - Choose an appropriate cost function
4. Design the model and train to find the optimal parameters of the model
   - Prepare a design matrix

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data
3. Formulate ML problem
   - Regression vs Classification
   - Choose an appropriate cost function
4. Design the model and train to find the optimal parameters of the model
   - Prepare a design matrix
   - Perform feature engineering

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data
3. Formulate ML problem
   - Regression vs Classification
   - Choose an appropriate cost function
4. Design the model and train to find the optimal parameters of the model
   - Prepare a design matrix
   - Perform feature engineering
   - Validate your choice of hyper-parameters using a cross-validation set

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data
3. Formulate ML problem
   - Regression vs Classification
   - Choose an appropriate cost function
4. Design the model and train to find the optimal parameters of the model
   - Prepare a design matrix
   - Perform feature engineering
   - Validate your choice of hyper-parameters using a cross-validation set
5. Evaluate the model on a test set

Machine Learning Problem Pipeline

1. Gather data
2. Visualize the data
3. Formulate ML problem
   - Regression vs Classification
   - Choose an appropriate cost function
4. Design the model and train to find the optimal parameters of the model
   - Prepare a design matrix
   - Perform feature engineering
   - Validate your choice of hyper-parameters using a cross-validation set
5. Evaluate the model on a test set
   - If the performance is not satisfactory, go back to step 4

Outline

NYU TANDON SCHOOL OF ENGINEERING

Course Outline

- Day 1: Intro to ML
- Day 2: Linear Regression
- Day 3: Generalization Error
- Day 4: Linear Classification
- Day 5: Mini-Project Competition & Presentations
- Day 6: Neural Networks
- Day 7: Deep Learning & Convolutional Neural Networks
- Day 8: Applications of CNNs
- Day 9: Final Projects
- Day 10: Final Projects & Presentations

NYU TANDON SCHOOL OF ENGINEERING

Course Format, Website, Resources

- Course Website: `github.com/nikopj/SummerML`

Course Format, Website, Resources

- Course Website: `github.com/nikopj/SummerML`
  - Github: share collections of documents, repositories of code

Course Format, Website, Resources

- Course Website: `github.com/nikopj/SummerML`
  - Github: share collections of documents, repositories of code
  - Contains lecture slides, code notebooks, and datasets

Course Format, Website, Resources

- Course Website: `github.com/nikopj/SummerML`
  - Github: share collections of documents, repositories of code
  - Contains lecture slides, code notebooks, and datasets
  - Slides posted before lecture, demo code and solutions posted after

Course Format, Website, Resources

- Course Website: `github.com/nikopj/SummerML`
  - Github: share collections of documents, repositories of code
  - Contains lecture slides, code notebooks, and datasets
  - Slides posted before lecture, demo code and solutions posted after
- We're strongly encourage programming in Python via Google Colab

Course Format, Website, Resources

- Course Website: `github.com/nikopj/SummerML`
  - Github: share collections of documents, repositories of code
  - Contains lecture slides, code notebooks, and datasets
  - Slides posted before lecture, demo code and solutions posted after
- We're strongly encourage programming in Python via Google Colab
  - No installation required

Course Format, Website, Resources

- Course Website: `github.com/nikopj/SummerML`
  - Github: share collections of documents, repositories of code
  - Contains lecture slides, code notebooks, and datasets
  - Slides posted before lecture, demo code and solutions posted after
- We're strongly encourage programming in Python via Google Colab
  - No installation required
- We'll give additional resources at the end of each day based on student interest

Outline

NYU TANDON SCHOOL OF ENGINEERING

Setting Up Python

- Scripting Python
  - Sequentially execute lines of code
- Jupyter Notebook
  - Code Blocks and Text Blocks (markdown)
  - Variables saved in workspace (across code blocks)
- Google Colab
  - Jupyter Notebook online (no real-time collaboration)
  - No installation
  - Free GPU for 12 hours
- Set up 1 of these 3 things and `print('hello world!')`

Outline

NYU TANDON SCHOOL OF ENGINEERING

Black-Box Abstraction

[¡+-¿]

- Can deal with many things with only knowing their **inputs** and **outputs**
- Need not know the details of the insides of our "boxes"
- Examples:
  - MTA turnstile
  - Python's `print()` function
  - Your Smart-Phone
  - Others?
- When does Black-Box Abstraction fail?

Outline

NYU TANDON SCHOOL OF ENGINEERING

Python Basics

- Program

Python Basics

- Program
  - We write operations to be executed on variables

Python Basics

- Program
  - We write operations to be executed on variables
- Variables

Python Basics

- Program
  - We write operations to be executed on variables
- Variables
  - Referencing and interacting with items in the program

Python Basics

- Program
  - We write operations to be executed on variables
- Variables
  - Referencing and interacting with items in the program
- If-Statements

Python Basics

- Program
  - We write operations to be executed on variables
- Variables
  - Referencing and interacting with items in the program
- If-Statements
  - Conditionally execute lines of code

Python Basics

- Program
  - We write operations to be executed on variables
- Variables
  - Referencing and interacting with items in the program
- If-Statements
  - Conditionally execute lines of code
- Functions

Python Basics

- Program
  - We write operations to be executed on variables
- Variables
  - Referencing and interacting with items in the program
- If-Statements
  - Conditionally execute lines of code
- Functions
  - Reuse lines of code at any time

Python Basics

- Lists

Python Basics

- Lists
  - Store an ordered collection of data

Python Basics

- Lists
  - Store an ordered collection of data
- Loops

Python Basics

- Lists
  - Store an ordered collection of data
- Loops
  - Conditionally re-execute code

Python Basics

- Lists
  - Store an ordered collection of data
- Loops
  - Conditionally re-execute code
- Strings

Python Basics

- Lists
  - Store an ordered collection of data
- Loops
  - Conditionally re-execute code
- Strings
  - Words and sentences are treated as lists of characters

Python Basics

- Lists
  - Store an ordered collection of data
- Loops
  - Conditionally re-execute code
- Strings
  - Words and sentences are treated as lists of characters
- Classes (advanced)

Python Basics

- Lists
  - Store an ordered collection of data
- Loops
  - Conditionally re-execute code
- Strings
  - Words and sentences are treated as lists of characters
- Classes (advanced)
  - Making your own data-type. Functions and variables made to be associated with it too.

Modules/Libraries/Packages

- NumPy: math, vectors and matrices
- MatPlotLib: plotting graphs, visualizing data
- Pandas: convenient for storing and retrieving data
- Sklearn: convenient wrapper for simple ML problems
- TensorFlow: computational graph, neural networks
- Keras: convenient wrapper for TF

Outline

NYU TANDON SCHOOL OF ENGINEERING

Vectors

- A **vector** is an n-tuple of numbers or symbols

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 5 + 6, 2 + 0, 9 + (-5)) = (4, 11, 2, 4)$

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 5 + 6, 2 + 0, 9 + (-5)) = (4, 11, 2, 4)$
- Vectors may be scaled by a number, element-wise

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 5 + 6, 2 + 0, 9 + (-5)) = (4, 11, 2, 4)$
- Vectors may be scaled by a number, element-wise
  - Ex: $3\mathbf{v} = (3 \times 1, 3 \times 5, 3 \times 2, 3 \times 9) = (3, 15, 6, 27)$

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 5 + 6, 2 + 0, 9 + (-5)) = (4, 11, 2, 4)$
- Vectors may be scaled by a number, element-wise
  - Ex: $3\mathbf{v} = (3 \times 1, 3 \times 5, 3 \times 2, 3 \times 9) = (3, 15, 6, 27)$
- **Dot-Product**: sum of element-wise products of two vectors

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 5 + 6, 2 + 0, 9 + (-5)) = (4, 11, 2, 4)$
- Vectors may be scaled by a number, element-wise
  - Ex: $3\mathbf{v} = (3 \times 1, 3 \times 5, 3 \times 2, 3 \times 9) = (3, 15, 6, 27)$
- **Dot-Product**: sum of element-wise products of two vectors
  - Ex:
    $\mathbf{u} \cdot \mathbf{v} = 3 \times 1 + 6 \times 5 + 0 \times 2 + (-5) \times 9 = 3 + 30 - 45 = -12$

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 5 + 6, 2 + 0, 9 + (-5)) = (4, 11, 2, 4)$
- Vectors may be scaled by a number, element-wise
  - Ex: $3\mathbf{v} = (3 \times 1, 3 \times 5, 3 \times 2, 3 \times 9) = (3, 15, 6, 27)$
- **Dot-Product**: sum of element-wise products of two vectors
  - Ex:
    $\mathbf{u} \cdot \mathbf{v} = 3 \times 1 + 6 \times 5 + 0 \times 2 + (-5) \times 9 = 3 + 30 - 45 = -12$
- Length of a Vector (L2 Norm) : $||\mathbf{x}|| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$

NYU TANDON SCHOOL OF ENGINEERING

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 5 + 6, 2 + 0, 9 + (-5)) = (4, 11, 2, 4)$
- Vectors may be scaled by a number, element-wise
  - Ex: $3\mathbf{v} = (3 \times 1, 3 \times 5, 3 \times 2, 3 \times 9) = (3, 15, 6, 27)$
- **Dot-Product**: sum of element-wise products of two vectors
  - Ex:
    $\mathbf{u} \cdot \mathbf{v} = 3 \times 1 + 6 \times 5 + 0 \times 2 + (-5) \times 9 = 3 + 30 - 45 = -12$
- Length of a Vector (L2 Norm) : $||\mathbf{x}|| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$
  - Ex: $||\mathbf{v}|| = \sqrt{1 + 25 + 4 + 81} = \sqrt{111}$

NYU TANDON SCHOOL OF ENGINEERING

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 5 + 6, 2 + 0, 9 + (-5)) = (4, 11, 2, 4)$
- Vectors may be scaled by a number, element-wise
  - Ex: $3\mathbf{v} = (3 \times 1, 3 \times 5, 3 \times 2, 3 \times 9) = (3, 15, 6, 27)$
- **Dot-Product**: sum of element-wise products of two vectors
  - Ex:
    $\mathbf{u} \cdot \mathbf{v} = 3 \times 1 + 6 \times 5 + 0 \times 2 + (-5) \times 9 = 3 + 30 - 45 = -12$
- Length of a Vector (L2 Norm) : $||\mathbf{x}|| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$
  - Ex: $||\mathbf{v}|| = \sqrt{1 + 25 + 4 + 81} = \sqrt{111}$
- Geometric Conceptualization of Vectors on board.

NYU TANDON SCHOOL OF ENGINEERING

Vectors

- A **vector** is an n-tuple of numbers or symbols
  - Ex: integer 4-tuple: $\mathbf{v} = (1, 5, 2, 9)$
- Vectors of the same size may be added together, element-wise
  - Ex: $\mathbf{u} = (3, 6, 0, -5)$
    $\mathbf{v} + \mathbf{u} = (1 + 3, 5 + 6, 2 + 0, 9 + (-5)) = (4, 11, 2, 4)$
- Vectors may be scaled by a number, element-wise
  - Ex: $3\mathbf{v} = (3 \times 1, 3 \times 5, 3 \times 2, 3 \times 9) = (3, 15, 6, 27)$
- **Dot-Product**: sum of element-wise products of two vectors
  - Ex:
    $\mathbf{u} \cdot \mathbf{v} = 3 \times 1 + 6 \times 5 + 0 \times 2 + (-5) \times 9 = 3 + 30 - 45 = -12$
- Length of a Vector (L2 Norm) : $||\mathbf{x}|| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$
  - Ex: $||\mathbf{v}|| = \sqrt{1 + 25 + 4 + 81} = \sqrt{111}$
- Geometric Conceptualization of Vectors on board.
  - What do we mean by **dimension**?

NYU TANDON SCHOOL OF ENGINEERING

Matrices

- A **matrix** is a rectangular array of numbers or symbols arranged in rows and columns

Matrices

- A **matrix** is a rectangular array of numbers or symbols arranged in rows and columns
  - Ex: 2 by 3 matrix, $M = \begin{bmatrix} 1 & 9 & -12 \\ 15 & -2 & 0 \end{bmatrix}$

Matrices

- A **matrix** is a rectangular array of numbers or symbols arranged in rows and columns
  - Ex: 2 by 3 matrix, $M = \begin{bmatrix} 1 & 9 & -12 \\ 15 & -2 & 0 \end{bmatrix}$
- Matrices of the same shape may be added together, element-wise

Matrices

- A **matrix** is a rectangular array of numbers or symbols arranged in rows and columns
  - Ex: 2 by 3 matrix, $M = \begin{bmatrix} 1 & 9 & -12 \\ 15 & -2 & 0 \end{bmatrix}$
- Matrices of the same shape may be added together, element-wise
  - Ex: $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 8 \\ 7 & 11 \end{bmatrix}, A + B = \begin{bmatrix} 1 & 9 \\ 9 & 12 \end{bmatrix}$

Matrices

- A **matrix** is a rectangular array of numbers or symbols arranged in rows and columns
  - Ex: 2 by 3 matrix, $M = \begin{bmatrix} 1 & 9 & -12 \\ 15 & -2 & 0 \end{bmatrix}$
- Matrices of the same shape may be added together, element-wise
  - Ex: $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 8 \\ 7 & 11 \end{bmatrix}, A + B = \begin{bmatrix} 1 & 9 \\ 9 & 12 \end{bmatrix}$
- Matrices may be scaled, element-wise

Matrices

- A **matrix** is a rectangular array of numbers or symbols arranged in rows and columns
  - Ex: 2 by 3 matrix, $M = \begin{bmatrix} 1 & 9 & -12 \\ 15 & -2 & 0 \end{bmatrix}$
- Matrices of the same shape may be added together, element-wise
  - Ex: $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 8 \\ 7 & 11 \end{bmatrix}, A + B = \begin{bmatrix} 1 & 9 \\ 9 & 12 \end{bmatrix}$
- Matrices may be scaled, element-wise
  - Ex: $aB = \begin{bmatrix} 0 & 8a \\ 7a & 11a \end{bmatrix}$, where a is a scalar

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria
  - $(C)_{ij} = c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}$

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria
  - $(C)_{ij} = c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}$
  - Criteria: for AB to be valid, # cols of A must equal the # rows of B

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria
  - $(C)_{ij} = c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}$
  - Criteria: for AB to be valid, # cols of A must equal the # rows of B
  - Result is a matrix with shape (# rows A, # cols B)

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria
  - $(C)_{ij} = c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}$
  - Criteria: for AB to be valid, # cols of A must equal the # rows of B
  - Result is a matrix with shape (# rows A, # cols B)
  - Ex: A is (2 x 3) and B is (3 x 5), $C = AB$ is (2 x 5)

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria
  - $(C)_{ij} = c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}$
  - Criteria: for AB to be valid, # cols of A must equal the # rows of B
  - Result is a matrix with shape (# rows A, # cols B)
  - Ex: A is (2 x 3) and B is (3 x 5), $C = AB$ is (2 x 5)
  - Example on board

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria
  - $(C)_{ij} = c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}$
  - Criteria: for AB to be valid, # cols of A must equal the # rows of B
  - Result is a matrix with shape (# rows A, # cols B)
  - Ex: A is (2 x 3) and B is (3 x 5), $C = AB$ is (2 x 5)
  - Example on board
- **Transpose**: A^T swaps the rows and columns of matrix A

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria
  - $(C)_{ij} = c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}$
  - Criteria: for AB to be valid, # cols of A must equal the # rows of B
  - Result is a matrix with shape (# rows A, # cols B)
  - Ex: A is (2 x 3) and B is (3 x 5), $C = AB$ is (2 x 5)
  - Example on board
- **Transpose**: A^T swaps the rows and columns of matrix A
- **Inverse**: A^{-1} satisfies the equation $AA^{-1} = A^{-1}A = \mathbb{I}$

More About Matrices

- Matrices may be multiplied together provided their shapes meet the criteria
  - $(C)_{ij} = c_{ij} = \sum_{k=1}^{N} a_{ik} b_{kj}$
  - Criteria: for AB to be valid, # cols of A must equal the # rows of B
  - Result is a matrix with shape (# rows A, # cols B)
  - Ex: A is (2 x 3) and B is (3 x 5), $C = AB$ is (2 x 5)
  - Example on board
- **Transpose**: A^T swaps the rows and columns of matrix A
- **Inverse**: A^{-1} satisfies the equation $AA^{-1} = A^{-1}A = \mathbb{I}$
  - Square matrices only!

Vectors and Matrices

- We may consider a vector as a matrix

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}$

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}$
  - **Column Vector**: shape (N x 1)

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}$
  - **Column Vector**: shape (N x 1)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 \\ 5 \\ 2 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}^T$

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}$
  - **Column Vector**: shape (N x 1)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 \\ 5 \\ 2 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}^T$
  - We'll consider vectors as column vectors by default

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}$
  - **Column Vector**: shape (N x 1)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 \\ 5 \\ 2 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}^T$
  - We'll consider vectors as column vectors by default
- Vector-Matrix Multiplication: on board

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}$
  - **Column Vector**: shape (N x 1)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 \\ 5 \\ 2 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}^T$
  - We'll consider vectors as column vectors by default
- Vector-Matrix Multiplication: on board
  - In what "dimension" is our vector starting, and where is it going to?

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}$
  - **Column Vector**: shape (N x 1)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 \\ 5 \\ 2 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}^T$
  - We'll consider vectors as column vectors by default
- Vector-Matrix Multiplication: on board
  - In what "dimension" is our vector starting, and where is it going to?
- Dot Product:

Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}$
  - **Column Vector**: shape (N x 1)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 \\ 5 \\ 2 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}^T$
  - We'll consider vectors as column vectors by default
- Vector-Matrix Multiplication: on board
  - In what "dimension" is our vector starting, and where is it going to?
- Dot Product:

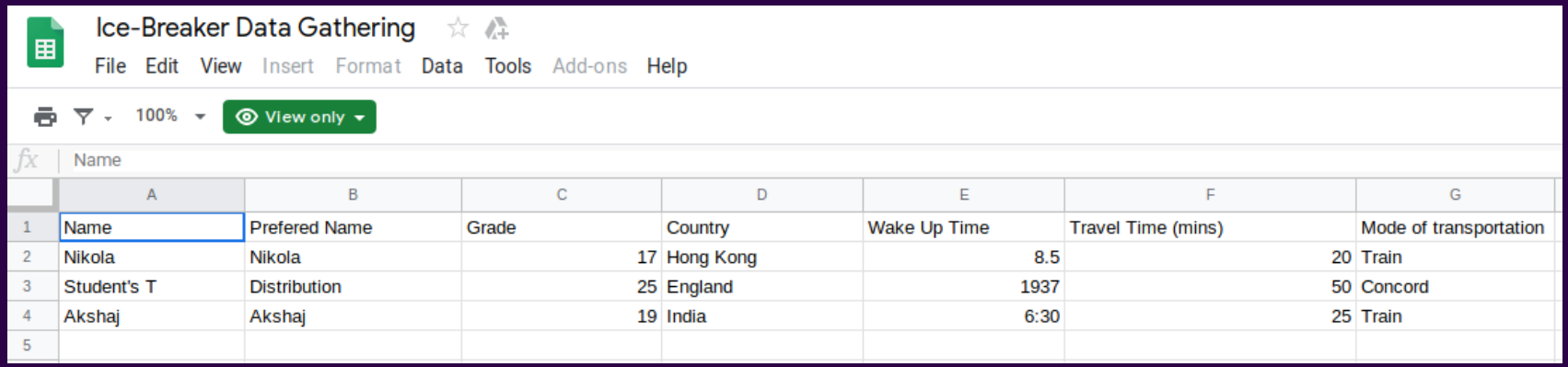
Vectors and Matrices

- We may consider a vector as a matrix
  - **Row Vector**: shape (1 x N)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}$
  - **Column Vector**: shape (N x 1)
    Ex: $\mathbf{v} = \begin{bmatrix} 1 \\ 5 \\ 2 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 2 & 9 \end{bmatrix}^T$
  - We'll consider vectors as column vectors by default
- Vector-Matrix Multiplication: on board
  - In what "dimension" is our vector starting, and where is it going to?
- Dot Product: $\mathbf{v}^\mathbf{T}\mathbf{v}$

Outline

Looking at our ice-breaker data in spreadsheets

- Columns have labels in the first row
- Collected data (numbers, words) follow below
- Let's export it to a Comma-Separated Values (CSV) file and open it



Ice-Breaker Data Gathering

File Edit View Insert Format Data Tools Add-ons Help

100% View only

fx Name

	A	B	C	D	E	F	G
1	Name	Prefered Name	Grade	Country	Wake Up Time	Travel Time (mins)	Mode of transportation
2	Nikola	Nikola	17	Hong Kong	8.5	20	Train
3	Student's T	Distribution	25	England	1937	50	Concord
4	Akshaj	Akshaj	19	India	6:30	25	Train
5							

Outline

NYU TANDON SCHOOL OF ENGINEERING

Lab: Plotting Functions

- Generate and plot the following functions in Python:
  - Scatter plot of points: (0,1), (2,3), (5,2), (4,1)
  - Straight Line: $y = mx + b$
  - Sine-wave $y = sin(x)$
  - Polynomial e.g. $y = x^3 + 2$
  - Exponential e.g. $y = e^-2x$
  - Gaussian (Use $\sigma = 0.5$)
  - Choose a function of your own
- Create separate plots for each of the functions, Compute the mean and variance of each function
- Use Wikipedia and Numpy Documentation to search for mathematical formulas and python functions

NYU TANDON SCHOOL OF ENGINEERING

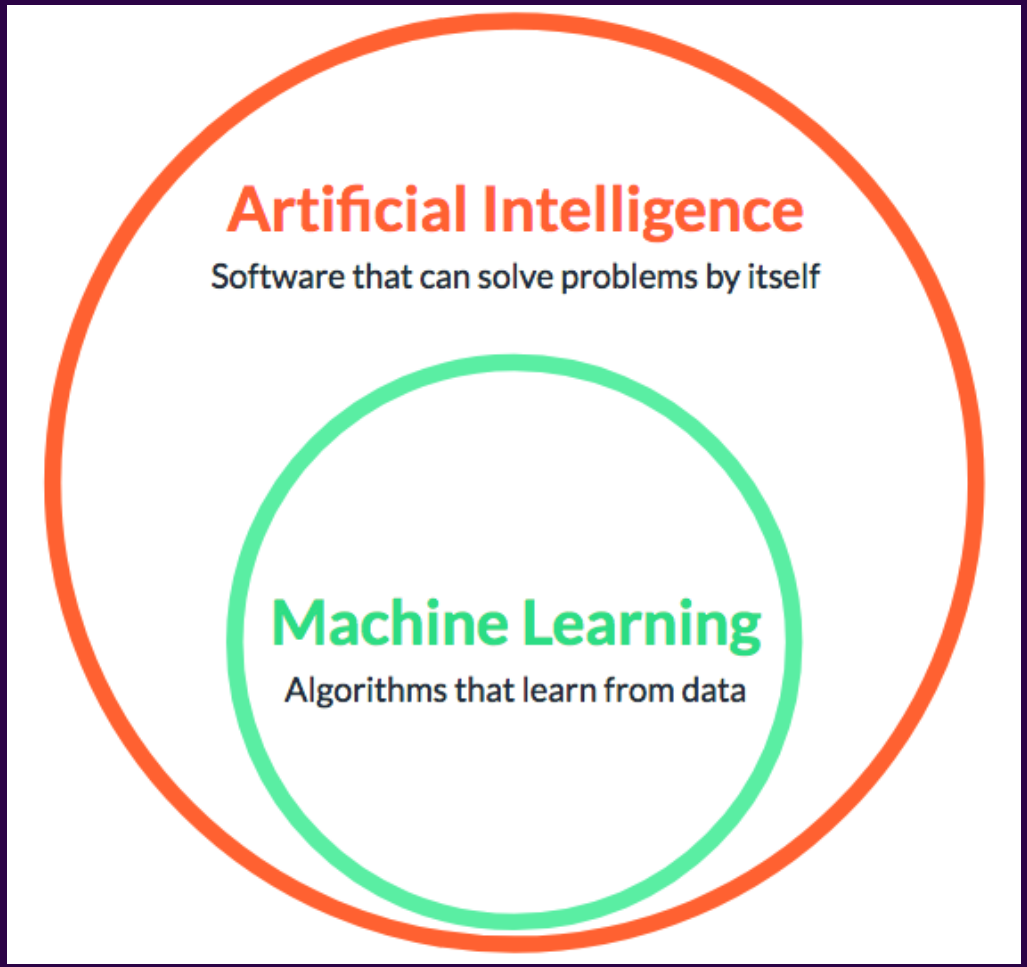
Outline

NYU TANDON SCHOOL OF ENGINEERING

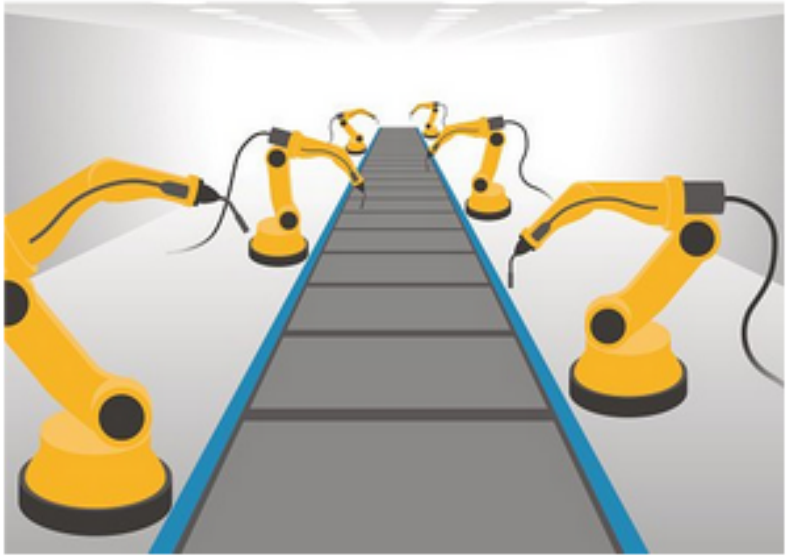
Artificial Intelligence

- Search
- Reasoning and Problem Solving
- Knowledge Representation
- Planning
- Learning
- Perception
- Natural Language Processing
- Motion and Manipulation
- Social and General Intelligence

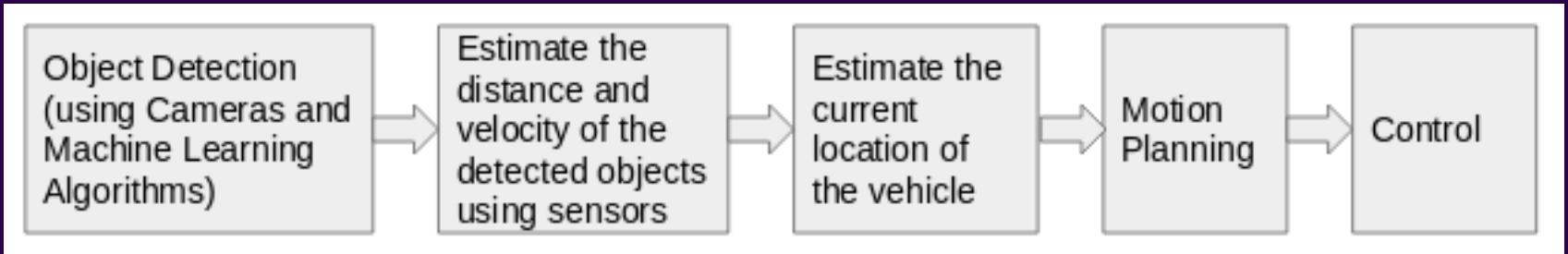
Machine Learning



Autonomous vs. Automated



Autonomous Example: Driver-less Cars



Why is Machine Learning so Prevalent?

- Database mining
- Medical records
- Computational biology
- Engineering
- Recommendation systems
- Understanding the human brain

Why Now?

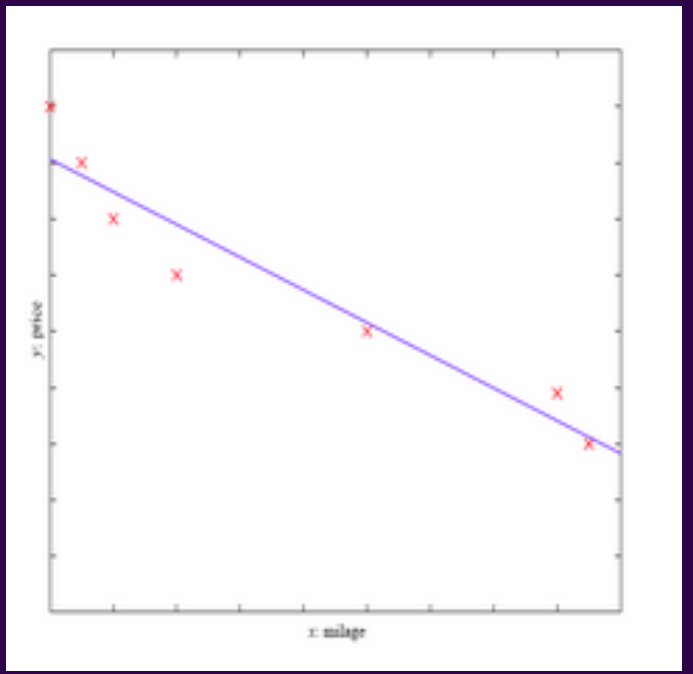
- Big Data
  - Massive storage. Large data centers
  - Massive connectivity
  - Sources of data from internet and elsewhere
- Computational advances
  - Distributed machines, clusters
  - GPUs and hardware

Outline

NYU TANDON SCHOOL OF ENGINEERING

What is Regression?

- Target variable is continuous-valued
- Example
  - Predict y = price of a car
  - From x = mileage, size, horsepower
  - Can use multiple predictors
- Assume some form of mapping
  - Ex: Linear mapping: $y = b + w_1 x$
  - Find parameter b, w_1 from data
- Use target-feature pairings as examples to form model



What is Classification?

- Determine what class a target belongs to based on its features
- Example:
  - Predict y = what type of object is in a photo
  - From x = the pixels of the image
- Learn a model/function from features to target
- Use target-feature pairings as examples to form model

Regression or Classification?

- **Problem 1:** Categorizing credit card applications into those who have good credit, bad credit and those who fall in the gray area.

Regression or Classification?

- **Problem 1:** Categorizing credit card applications into those who have good credit, bad credit and those who fall in the gray area.
  - Classification Problem

Regression or Classification?

- **Problem 1:** Categorizing credit card applications into those who have good credit, bad credit and those who fall in the gray area.
  - Classification Problem
  - Class Labels: Good credit, Bad credit, Average credit

Regression or Classification?

- **Problem 1:** Categorizing credit card applications into those who have good credit, bad credit and those who fall in the gray area.
  - Classification Problem
  - Class Labels: Good credit, Bad credit, Average credit
- **Problem 2:** Determining the sentiment of customer reviews for a product on Amazon.

Regression or Classification?

- **Problem 1:** Categorizing credit card applications into those who have good credit, bad credit and those who fall in the gray area.
  - Classification Problem
  - Class Labels: Good credit, Bad credit, Average credit
- **Problem 2:** Determining the sentiment of customer reviews for a product on Amazon.
  - Classification Problem

Regression or Classification?

- **Problem 1:** Categorizing credit card applications into those who have good credit, bad credit and those who fall in the gray area.
  - Classification Problem
  - Class Labels: Good credit, Bad credit, Average credit
- **Problem 2:** Determining the sentiment of customer reviews for a product on Amazon.
  - Classification Problem
  - Class Labels: Positive, Negative, Neutral

Regression or Classification?

- **Problem 1:** Categorizing credit card applications into those who have good credit, bad credit and those who fall in the gray area.
  - Classification Problem
  - Class Labels: Good credit, Bad credit, Average credit
- **Problem 2:** Determining the sentiment of customer reviews for a product on Amazon.
  - Classification Problem
  - Class Labels: Positive, Negative, Neutral
- **Problem 3:** Predict whether an employee's income is over 100k a year or not.

Regression or Classification?

- **Problem 1:** Categorizing credit card applications into those who have good credit, bad credit and those who fall in the gray area.
  - Classification Problem
  - Class Labels: Good credit, Bad credit, Average credit
- **Problem 2:** Determining the sentiment of customer reviews for a product on Amazon.
  - Classification Problem
  - Class Labels: Positive, Negative, Neutral
- **Problem 3:** Predict whether an employee's income is over 100k a year or not.
  - Classification Problem

Regression or Classification?

- **Problem 1:** Categorizing credit card applications into those who have good credit, bad credit and those who fall in the gray area.
  - Classification Problem
  - Class Labels: Good credit, Bad credit, Average credit
- **Problem 2:** Determining the sentiment of customer reviews for a product on Amazon.
  - Classification Problem
  - Class Labels: Positive, Negative, Neutral
- **Problem 3:** Predict whether an employee's income is over 100k a year or not.
  - Classification Problem
  - Class Labels: Over 100k, Under 100k

Regression or Classification?

- **Problem 4:** Estimating change in climate.

Regression or Classification?

- **Problem 4:** Estimating change in climate.
  - Regression Problem

Regression or Classification?

- **Problem 4:** Estimating change in climate.
  - Regression Problem
  - Target Variable: Predicting future temperatures

Regression or Classification?

- **Problem 4:** Estimating change in climate.
  - Regression Problem
  - Target Variable: Predicting future temperatures
- **Problem 5:** Identifying hate speech in social media.

Regression or Classification?

- **Problem 4:** Estimating change in climate.
  - Regression Problem
  - Target Variable: Predicting future temperatures
- **Problem 5:** Identifying hate speech in social media.
  - Classification Problem

Regression or Classification?

- **Problem 4:** Estimating change in climate.
  - Regression Problem
  - Target Variable: Predicting future temperatures
- **Problem 5:** Identifying hate speech in social media.
  - Classification Problem
  - Class labels: Normal Speech, Hate Speech

Regression or Classification?

- **Problem 4:** Estimating change in climate.
  - Regression Problem
  - Target Variable: Predicting future temperatures
- **Problem 5:** Identifying hate speech in social media.
  - Classification Problem
  - Class labels: Normal Speech, Hate Speech
- **Problem 6:** Forecasting the energy demand in a region.

Regression or Classification?

- **Problem 4:** Estimating change in climate.
  - Regression Problem
  - Target Variable: Predicting future temperatures
- **Problem 5:** Identifying hate speech in social media.
  - Classification Problem
  - Class labels: Normal Speech, Hate Speech
- **Problem 6:** Forecasting the energy demand in a region.
  - Regression Problem

Regression or Classification?

- **Problem 4:** Estimating change in climate.
  - Regression Problem
  - Target Variable: Predicting future temperatures
- **Problem 5:** Identifying hate speech in social media.
  - Classification Problem
  - Class labels: Normal Speech, Hate Speech
- **Problem 6:** Forecasting the energy demand in a region.
  - Regression Problem
  - Target Variable: Predicting the amount of energy needed in the future

Thank You!

- Next Class: ...