Machine Learning **Experimentation with Go**
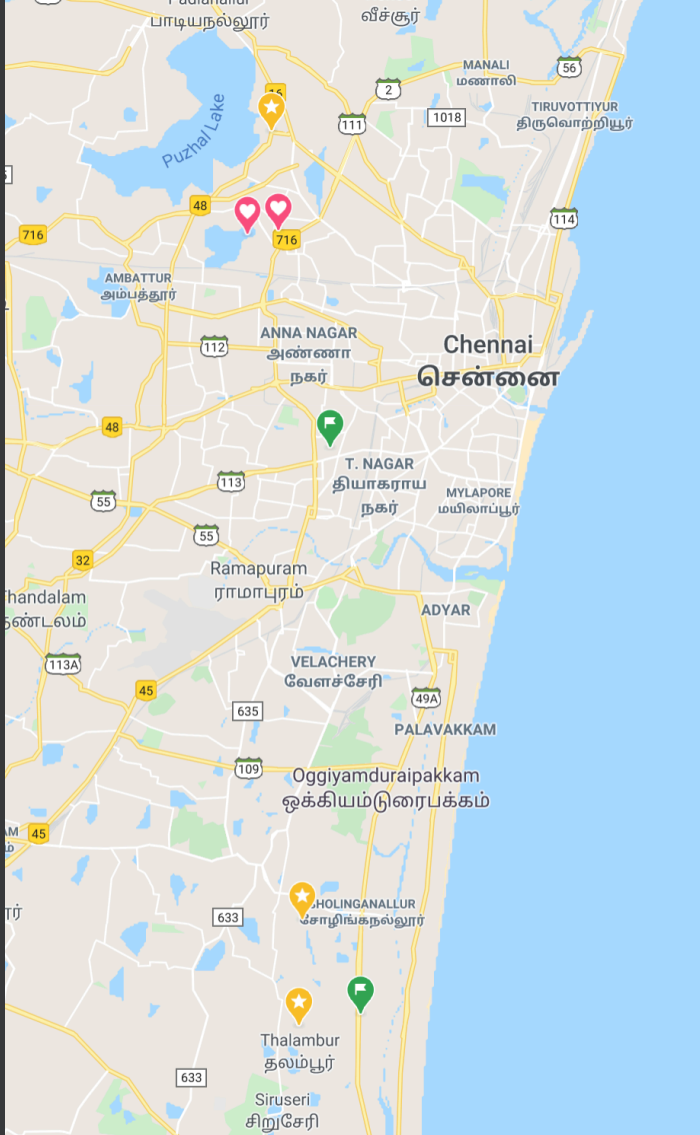
GK Senthil

But why?

For today

Machine Learning – an overview

Solving problems with Machine Learning

Where Go fits in..

More to explore..

Renting a house in Chennai

2 BHK	1000 sqft	Furnished	Adyar	Rs. 25000
2 BHK	550 sqft	Furnished	Adyar	Rs. 16000
3 BHK	1750 sqft	Semi furnished	Sholinganallur	Rs. 22000
2 BHK	1500 sqft	Semi-furnished	Thoraippakkam	?

Machine Learning

Input
+
Rules

= Outcomes

Programming

Input
+
Outcomes

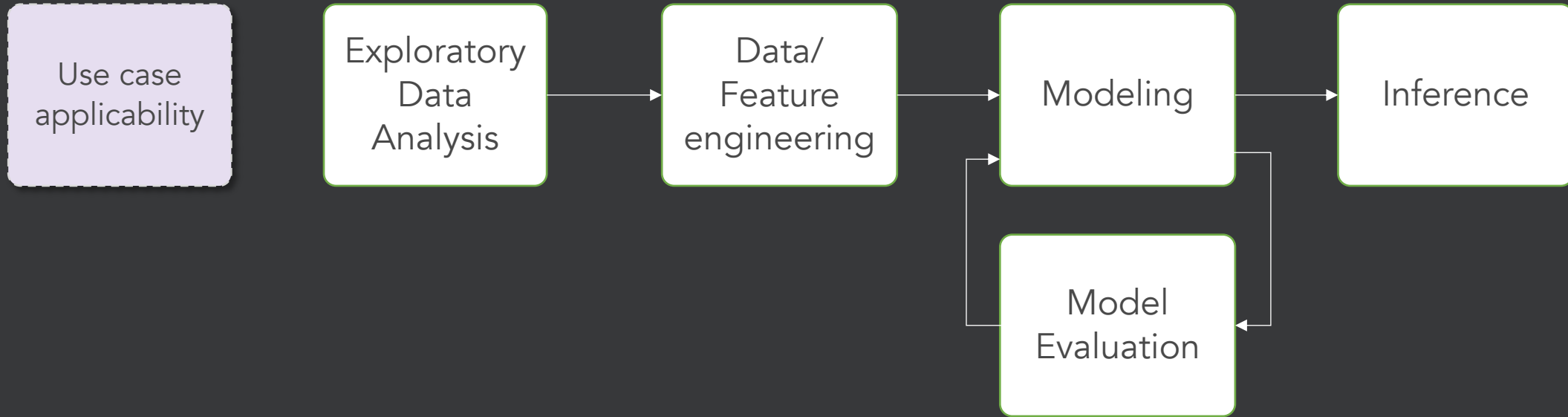+General rules (**Algorithms**)

~ Rules

(Machine) Learning

Input ~ Training data
Outcomes ~ Labels

So many ways machines can learn..

- Supervised
  - learning based on known outcomes

- Unsupervised
  - finding relationships in the data

- Reinforcement learning
  - Maximizing reward based on signal

- "Traditional" Machine learning
  - Requires structured data as "features" for learning

- Deep learning/Neural networks
  - Algorithm learns to "learn" features also works for unstructured data

Solving with Machine Learning (Structured data)

Dataset

https://archive.ics.uci.edu/ml/datasets.php

https://www.kaggle.com/datasets

Data portal: Open Data monitor –

https://opendatamonitor.eu/frontend/web/index.php?r=dashboard%2Findex

Exploratory Data Analysis

- Understanding Attribute data types

- Presence of values and their range

- Statistics of attributes – Mean, Variance etc.

- Distribution of attributes and target (Histogram)

- Correlation among attributes

Data/Feature Engineering

- Fill missing data

- Converting "String" categories/qualifiers to numeric values

- Combining features

- Dropping unnecessary features

- *Dimensionality reduction*

- And many more..

"Learning" – Linear Regression (Ordinary Least Squares)

- Model Function:
  - Sum of weighted inputs of features + Bias

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Objective : Find values for weights such that the Root Mean Square Error is a minimum

$$\text{MSE}\left(\mathbf{X}, h_{\boldsymbol{\theta}}\right) = \frac{1}{m} \sum_{i=1}^{m} \left(\boldsymbol{\theta}^\intercal \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

- x is the "vector" of features for one given training data sample
- Ø is the "vector" of weights
- y is the target value
- m is the number of training data samples

Solving for "Theta"

Option 1 – Normal Function

$$\hat{\boldsymbol{\theta}} = \left(\mathbf{X}^\intercal \mathbf{X}\right)^{-1} \mathbf{X}^\intercal \mathbf{y}$$

Option 2 – Gradient Descent
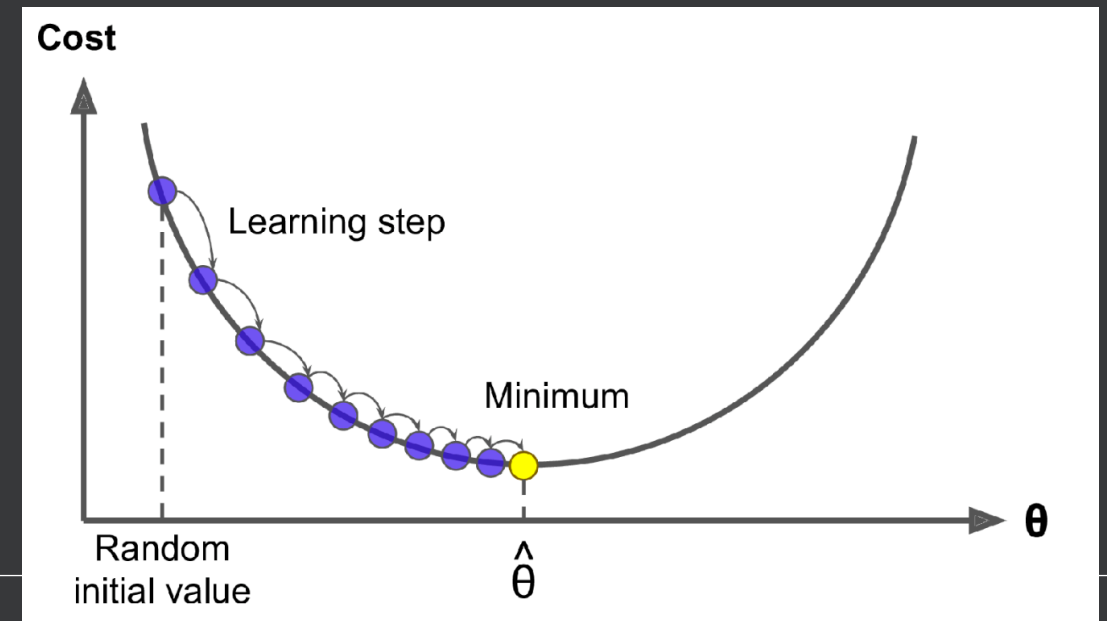
Inference

- Execute the model "function" for a given input

How Go fits in:
EDA &
Feature engineering

[iPython]

Reading a CSV file and displaying a column

```python
dayRentals = pd.read_csv("./day.csv")

dayRentals.head()

dayRentals["mnth"].value_counts()
```

Plotting a graph

```python
plt.scatter(x, y)

plt.show()
```

```go
package main
import "fmt"
import "encoding/csv"
import "os"
import "strconv"

func main(){
    // Open the CSV.
    f, err := os.Open("day.csv")
    // Read in the CSV records.
    r := csv.NewReader(f)
    records, err := r.ReadAll()
    if err != nil{
        fmt.Println(err)
    }
    var i = 0
    var month_counts[13] int
    for _, record := range records {
        i++
        if (i<=5){
            fmt.Println(record)
        }
        month, err := strconv.Atoi (record[4])
        if (err != nil){

        }
        month_counts[month] += 1
    }
    for i := 1; i < 13; i++ {
        fmt.Println("Month: "+strconv.Itoa(i)+" "+strconv
    }
}
```

```go
func main() {
    rand.Seed(int64(0))

    p, err := plot.New()
    if err != nil {
        panic(err)
    }

    p.Title.Text = "Plotutil example"
    p.X.Label.Text = "X"
    p.Y.Label.Text = "Y"

    err = plotutil.AddLinePoints(p,
        "First", randomPoints(15),
        "Second", randomPoints(15),
        "Third", randomPoints(15))
    if err != nil {
        panic(err)
    }

    // Save the plot to a PNG file.
    if err := p.Save(4*vg.Inch, 4*vg.Inch, "points.png"); err != nil {
        panic(err)
    }
}

// randomPoints returns some random x, y points.
func randomPoints(n int) plotter.XYs {
    pts := make(plotter.XYs, n)
    for i := range pts {
        if i == 0 {
            pts[i].X = rand.Float64()
        } else {
            pts[i].X = pts[i-1].X + rand.Float64()
        }
        pts[i].Y = pts[i].X + 10*rand.Float64()
    }
    return pts
}
```
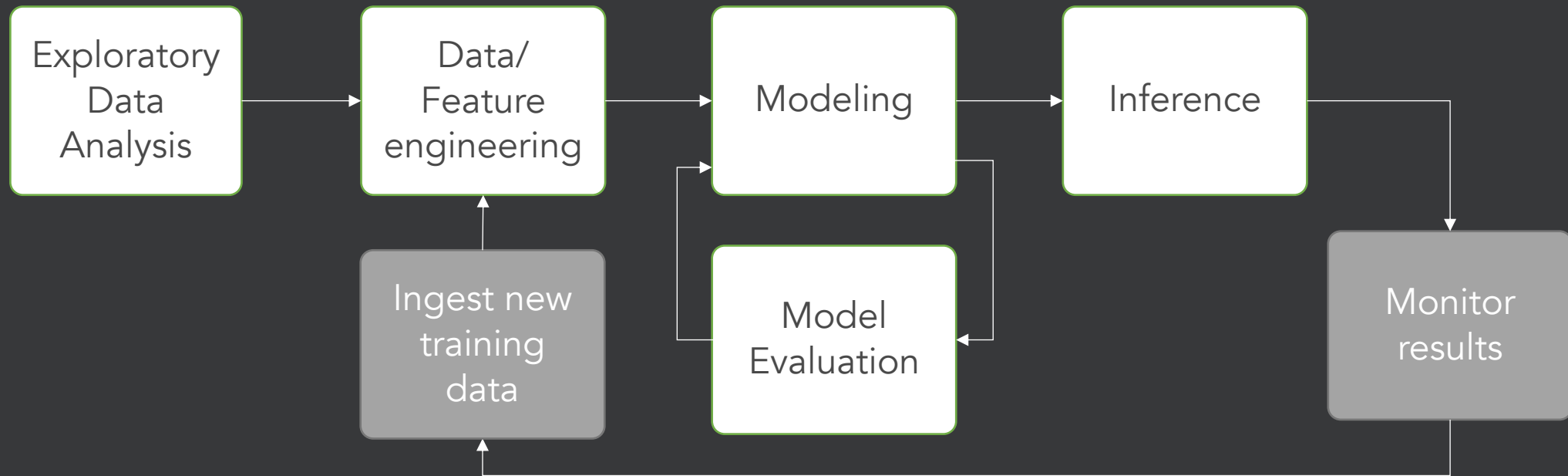
Modeling with Go

Deep Learning

- gorgonia.org/gorgonia

- Computer vision
  https://github.com/hybridgroup/gocv

- Tensorflow bindings

To be further explored

Pipeline

```
┌─────────────┐      ┌─────────────┐      ┌─────────────┐      ┌─────────────┐
│ Exploratory │ ───► │   Data/     │ ───► │  Modeling   │ ───► │  Inference  │
│    Data     │      │  Feature    │      │             │      │             │
│  Analysis   │      │ engineering │      │             │      │             │
└─────────────┘      └─────────────┘      └─────────────┘      └─────────────┘
                           ▲                    ▲  │
                           │                    │  ▼
                     ┌─────────────┐      ┌─────────────┐      ┌─────────────┐
                     │ Ingest new  │      │   Model     │      │  Monitor    │
                     │  training   │      │ Evaluation  │      │  results    │
                     │    data     │      │             │      │             │
                     └─────────────┘      └─────────────┘      └─────────────┘
```

Static typing	Performance	Language familiarity

Some libraries..

Dataframe
- github.com/go-gota/gota/dataframe

Graphs
- github.com/gonum.org/v1/plot/...

Vector/Matrix operations
- https://github.com/gonum/gonum

Go Jupyter notebook binding
https://github.com/gopherdata/gophernotes

Linear regression - github.com/sajari/regression

Golearn- github.com/sjwhitworth/golearn
Implements several classification algorithms and basic neural nets

Machine Learning With Go - Second Edition

Daniel Whitenack, Janani Selvaraj

Announcing Meetup Group Techcelerator!

Thank you!