COS30019 – Intro to AI
Introduction to Machine Learning (ML)

1

Outline

- What is ML?
  - ML – Why, What and When?
- Types of Machine Learning
  - Supervised learning
  - Unsupervised learning
  - Semi-supervised learning
  - Reinforcement learning
- Supervised Learning with Linear Regression
- Designing a Learning System

2

ML – Why?

- Big data
  - Large datasets from growth of automation/web.
    E.g., Web click data, medical records, biology, engineering
- Applications can't program by hand.
  - E.g., Autonomous helicopter, handwriting recognition, most of Natural Language Processing (NLP), Computer Vision.
- Self-customizing programs
  - E.g., Amazon, Netflix product recommendations
- Understanding human learning (brain, real AI).

3

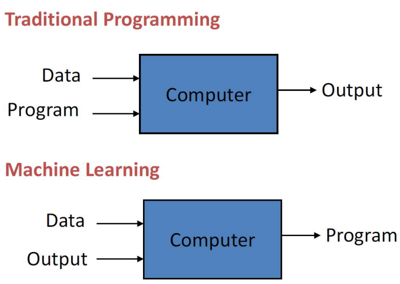
ML – What?

- "Learning is any process by which a system improves performance from experience." - Herbert Simon

- "Learning Problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E." - Tom Mitchell (1998)

4

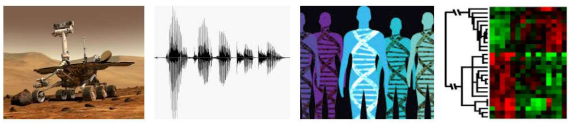
ML vs Traditional programming



5

ML – When to Use Machine Learning??

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



Learning isn't always useful:
- There is no need to "learn" to calculate payroll

6

A classic example of a task that requires machine learning:
It is very hard to say what makes a 2

Slide credit: Geoffrey Hinton

7

ML – Examples

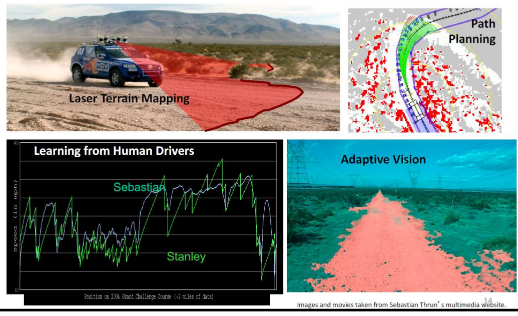
- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
  - Future stock prices or currency exchange rates

8

State of the Art Applications of Machine Learning

9

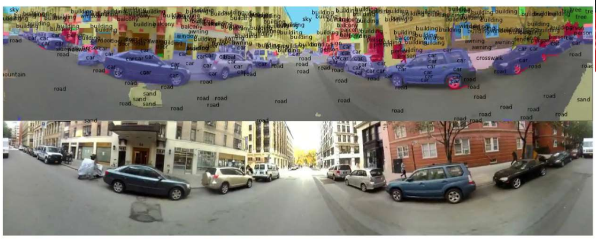
Autonomous Car Technology



Images and movies taken from Sebastian Thrun's multimedia website.

10

Scene Labeling via Deep Learning

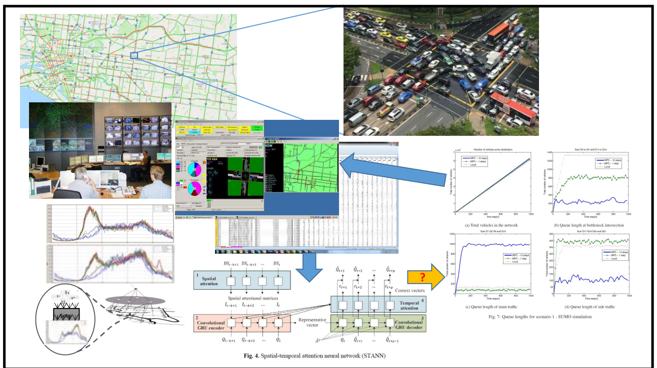


[Farabet et al. ICML 2012, PAMI 2013]

11

AI & ML @Swinburne

12



13

Types of Machine Learning

14

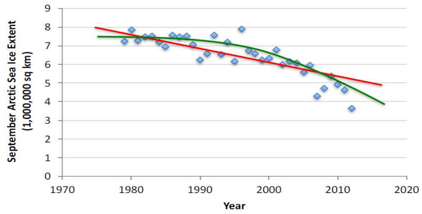
Types of Learning

- Supervised (inductive) learning
  - Given: training data + desired outputs (labels)
- Unsupervised learning
  - Given: training data (without desired outputs)
- Semi-supervised learning
  - Given: training data + a few desired outputs
- Reinforcement learning
  - Rewards from sequence of actions

15

Supervised Learning: Regression

Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is real-valued == regression

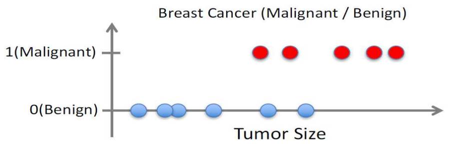


Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

16

Supervised Learning: Classification

Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification

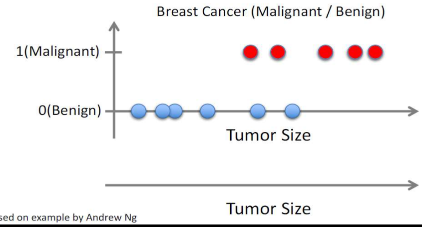


Based on example by Andrew Ng

17

Supervised Learning: Classification

Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification



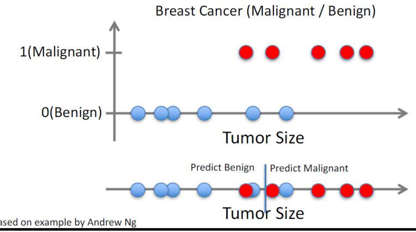
Based on example by Andrew Ng

18

Supervised Learning: Classification

Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x

– y is categorical == classification



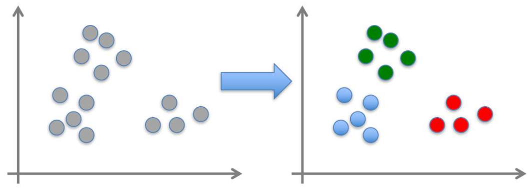
Based on example by Andrew Ng

19

Unsupervised Learning

Given $x_1, x_2, \ldots, x_n$ (without labels)
- Output hidden structure behind the x's

– E.g., clustering



E.g., https://news.google.com uses clustering

20

Reinforcement Learning

Given a sequence of states and actions with (delayed) rewards, output a policy

– Policy is a mapping from states → actions that tells you what to do in a given state

Examples:

– Credit assignment problem

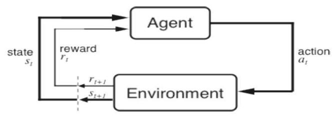
– Game playing

– Robot in a maze

– Balance a pole on your hand

21

The Agent-Environment Interface



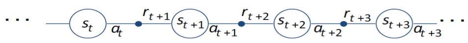
Agent and environment interact at discrete time steps : $t = 0, 1, 2, K$

Agent observes state at step t: $s_t \in S$

produces action at step t: $a_t \in A(s_t)$

gets resulting reward : $r_{t+1} \in \Re$

and resulting next state : s_{t+1}



Slide credit: Sutton & Barto

22

Reinforcement Learning

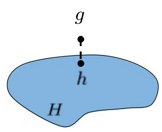
Learn policy from user demonstrations

Stanford Autonomous Helicopter
http://heli.stanford.edu/

23

Inductive Learning (Science)

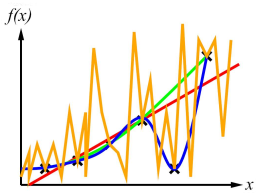
- Simplest form: learn a function from examples
  - A target function: g
  - Examples: input-output pairs $(x, g(x))$
  - E.g. x is an email and $g(x)$ is spam / ham
  - E.g. x is a house and $g(x)$ is its selling price
- Problem:
  - Given a hypothesis space H
  - Given a training set of examples x_i
  - Find a hypothesis $h(x)$ such that $h \sim g$
- Includes:
  - Classification (outputs = class labels)
  - Regression (outputs = real numbers)



24

Inductive Learning

- Curve fitting (regression, function approximation):



- Consistency vs. simplicity
- Ockham's razor

25

Supervised Learning with Linear Regression

26

Linear Regression (LR)

- A statistical regression method used for predictive analysis
- LR shows the linear relationship between the independent (predictor) variable i.e. X-axis and the dependent (output) variable i.e. Y-axis

Source: https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/

27

Linear Regression (LR)

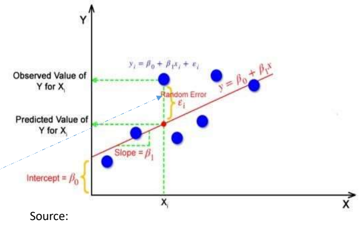
- How to compute the **best-fit line**?

  $y = h_\beta(x) = \beta_0 + \beta_1 x$, where:

  - y = Dependent variable,
  - x = Independent variable,
  - β_0 = constant/Intercept,
  - β_1 = Slope.

For each training example (x_i, y_i), we can compute the *random error* (or, *residual*) $\varepsilon_i = y_{predicted} - y_i$

where $y_{predicted} = h_\beta(x_i) = \beta_0 + \beta_1 x_i$



Source: https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/

28

Linear Regression (LR) - Best-fit Line

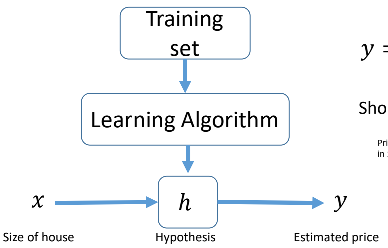
- What is the best fit line, i.e., what are the best values for β_0 and β_1?
- **Cost Function**:
  - Measuring the error a given line $y = h_\beta(x) = \beta_0 + \beta_1 x$ has with respect to a set of training examples {(x_1, y_1), …., (x_n, y_n)}
  - For instance, the **Mean Squared Error (MSE)** cost function:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i))^2$$

  - The line $y = h_\beta(x) = \beta_0 + \beta_1 x$ is "best fit" if MSE is minimised

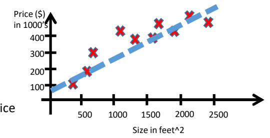
29

Model representation



$$y = h_\beta(x) = \beta_0 + \beta_1 x$$

Shorthand $h(x)$



Univariate linear regression

Slide credit: Andrew Ng

30

- **Hypothesis:** $h_\beta(x) = \beta_0 + \beta_1 x$
- **Parameters:** β_0, β_1
- **Cost function:** $J(\beta_0, \beta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(h_\beta(x_i) - y_i\right)^2$
- **Goal:** $\underset{\beta_0, \beta_1}{\text{minimize}}\ J(\beta_0, \beta_1)$

Slide credit: Andrew Ng

31

Gradient descent

Have some function $J(\beta_0, \beta_1)$

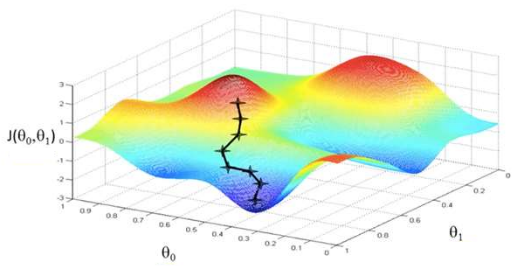
Want $\underset{\beta_0, \beta_1}{\text{argmin}}\ J(\beta_0, \beta_1)$

Outline:

- Start with some β_0, β_1
- Keep changing β_0, β_1 to reduce $J(\beta_0, \beta_1)$ until we hopefully end up at minimum

Slide credit: Andrew Ng

32



Slide credit: Andrew Ng

33

Gradient descent

Repeat until convergence {

$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\beta_0, \beta_1)$ (for $j = 0$ and $j = 1$)

}

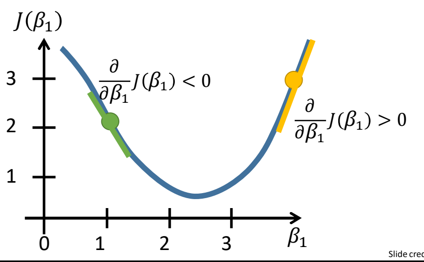
α: Learning rate (step size)

$\frac{\partial}{\partial \beta_j} J(\beta_0, \beta_1)$: derivative (rate of change)

Slide credit: Andrew Ng

34

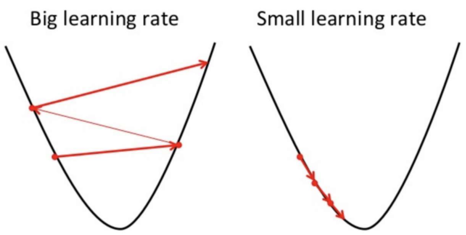
$$\beta_1 := \beta_1 - \alpha \frac{\partial}{\partial \beta_1} J(\beta_1)$$



Slide credit: Andrew Ng

35

Learning rate



36

Gradient descent for linear regression

Repeat until convergence{

$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\beta_0, \beta_1)$ (for $j = 0$ and $j = 1$)

}

- Linear regression model

$h_\beta(x) = \beta_0 + \beta_1 x$

$$J(\beta_0, \beta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(h_\beta(x_i) - y_i\right)^2$$

Slide credit: Andrew Ng

37

Computing partial derivative

- $\frac{\partial}{\partial \beta_j} J(\beta_0, \beta_1) = \frac{\partial}{\partial \beta_j} \frac{1}{n} \sum_{i=1}^{n}\left(h_\beta(x_i) - y_i\right)^2$
  $= \frac{\partial}{\partial \beta_j} \frac{1}{n} \sum_{i=1}^{n}(\beta_0 + \beta_1 x_i - y_i)^2$

- $j = 0$: $\frac{\partial}{\partial \beta_0} J(\beta_0, \beta_1) = \frac{2}{n} \sum_{i=1}^{n}\left(h_\beta(x_i) - y_i\right)$
- $j = 1$: $\frac{\partial}{\partial \beta_1} J(\beta_0, \beta_1) = \frac{2}{n} \sum_{i=1}^{n}\left(h_\beta(x_i) - y_i\right) x_i$

Slide credit: Andrew Ng

38

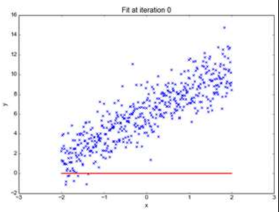
Gradient descent for linear regression

Repeat until convergence {

$$\beta_0 := \beta_0 - \alpha \frac{2}{n}\sum_{i=1}^{n}\left(h_\beta(x_i) - y_i\right)$$

$$\beta_1 := \beta_1 - \alpha \frac{2}{n}\sum_{i=1}^{n}\left(h_\beta(x_i) - y_i\right) x_i$$

}



Update β_0 and β_1 simultaneously

Slide credit: Andrew Ng

39

Regression: How to

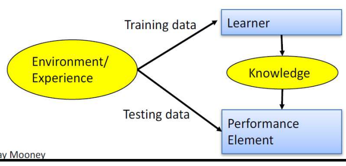
- So, do I have to write a program to calculate the "right" values of a and b to "minimize" the errors?
- And, how about multi-dimensional datasets?

- Short answer: Just use an existing library!
  - E.g., scikit-learn (see: https://scikit-learn.org/stable/)
  - E.g., tensorflow (see: https://www.tensorflow.org/)

40

Designing a Learning System

- Choose the training experience
- Choose exactly what is to be learned – i.e. the ***target function***
- Choose how to represent the target function
- Choose a learning algorithm to infer the target function from the experience



Based on slide by Ray Mooney

41

ML in a Nutshell

- Tens of thousands of machine learning algorithms
  - Hundreds new every year
- Every ML algorithm has three components:
  - **Representation**
  - **Optimization**
  - **Evaluation**

42

Various Function Representations

- Numerical functions
  - Linear regression
  - Neural networks
  - Support vector machines
- Symbolic functions
  - Decision trees
  - Rules in propositional logic
  - Rules in first-order predicate logic
- Instance-based functions
  - Nearest-neighbor
  - Case-based
- Probabilistic Graphical Models
  - Naïve Bayes
  - Bayesian networks
  - Hidden-Markov Models (HMMs)
  - Probabilistic Context Free Grammars (PCFGs)
  - Markov networks

43

Various Search/Optimization Algorithms

- Gradient descent
  - Perceptron
  - Backpropagation
- Dynamic Programming
  - HMM Learning
  - PCFG Learning
- Divide and Conquer
  - Decision tree induction
  - Rule learning
- Evolutionary Computation
  - Genetic Algorithms (GAs)
  - Genetic Programming (GP)
  - Neuro-evolution

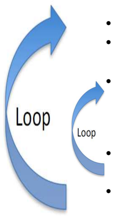
44

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- etc.

45

ML in Practice



- Understand domain, prior knowledge, and goals
- Data gathering, data integration, selection, cleaning, pre-processing, etc.
- Learn models
  - Model Selection
  - Model Training
  - Model Evaluation
- Interpret results
  - Hyperparameter tuning
- Consolidate and deploy discovered knowledge

46

Summary

- Learning can be viewed as using direct or indirect experience to approximate a chosen target function.
- Function approximation can be viewed as a search through a space of hypotheses (representations of functions) for one that best fits a set of training data.
- Different learning methods assume different hypothesis spaces (representation languages) and/or employ different search techniques.

47

ML for a practitioner

- If you:
  - Are a beginner
  - Need quick results
  - Dataset is simple (small and well-structured, e.g. a CSV file)
- Then consider Weka (https://www.cs.waikato.ac.nz/ml/weka/)
  - GUI
  - Easy to use

48

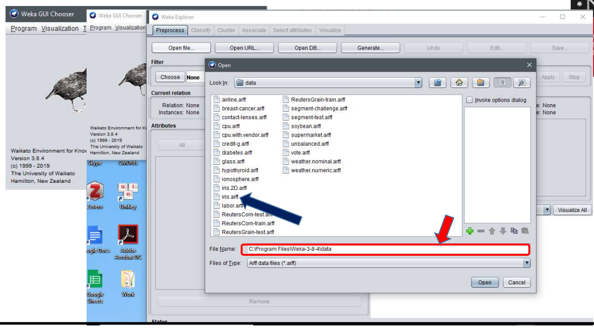
Some resources to get started with WEKA

https://www.analyticsvidhya.com/learning-paths-data-science-business-analytics-business-intelligence-big-data/weka-gui-learn-machine-learning/

https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf

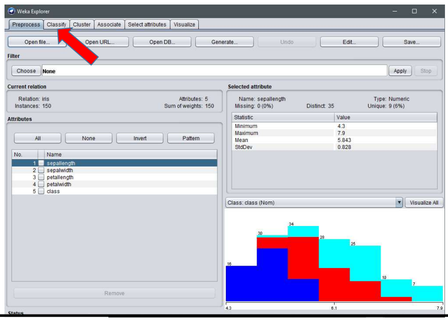
49

WEKA



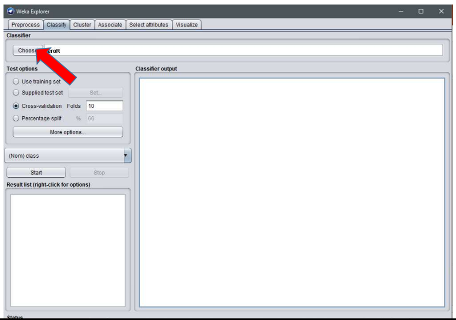
50

WEKA



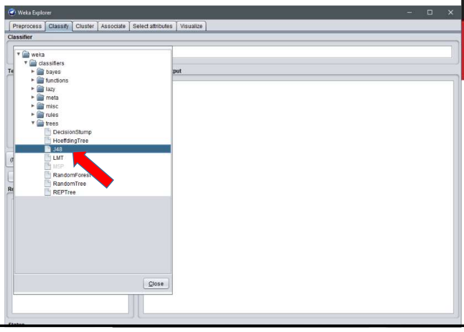
51

WEKA



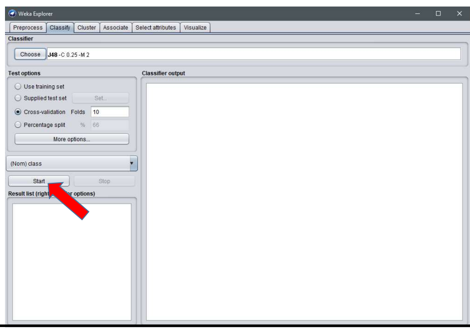
52

WEKA



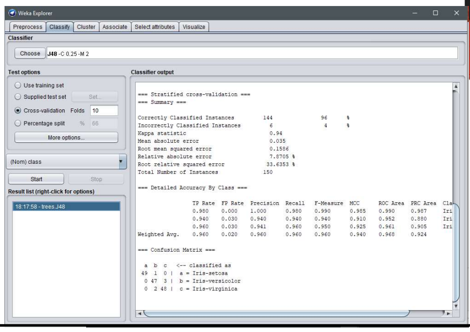
53

WEKA



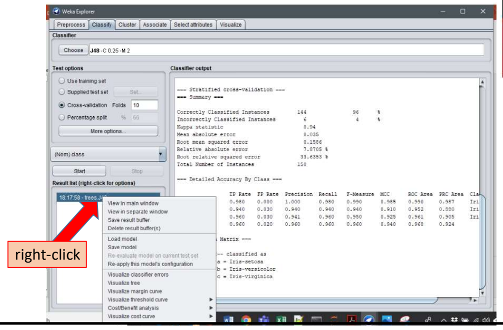
54

WEKA



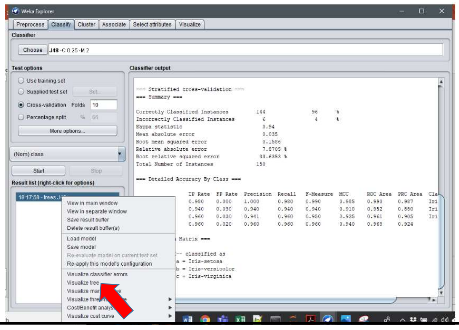
55

WEKA



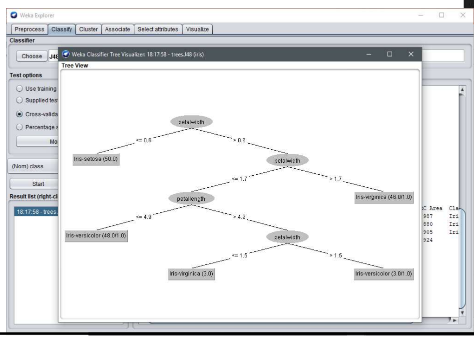
56

WEKA



57

WEKA

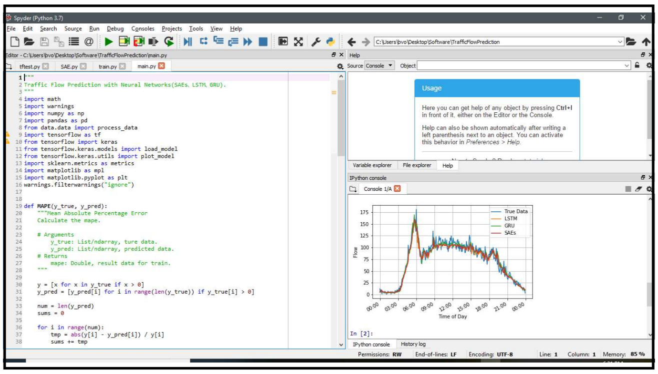


58

ML for a practitioner

- If you:
  - Are really serious about ML
  - Are ready to spend substantial amount of time to learn
  - Datasets can be really complex/noisy
- Then it's time to move on to a more serious tools:
  - R, python
  - Needs programming, application integration, libraries (tensorflow, pytorch, numpy, scikit-learn, etc.)!

59



60