Objective

- Introduction to basic developer concepts in Machine Learning
- Demonstrate with a Natural Language Text Classification example
Machine Learning (ML) in a Nutshell

- Broad topic covering the practices required to:
  - Ask a question that can be answered from data
  - Prepare selected data sources for machine processing
  - Apply ML processing to produce models from data
  - Use these models to predict or infer various outcomes based on new data

- Based on methods developed in the discipline of Data Science
  - Numerical techniques and algorithms for analyzing data to predict characteristics of new data
  - Techniques and algorithms sourced from different disciplines
Context – traditional development processes

- Developers map business problems to algorithms via programming language constructs
  - We define Class entities
  - We define frameworks built on design patterns that orchestrate how objects can interact with other objects
  - And so on….
- The developer maps real-world problems to these concepts and designs a new application
- This has worked and has been improved for a long time…
- But it is limited to a class of problems.
Context – ML Development Processes

- *Machine Learning is different*
- Business problems are solved by focusing on how to harvest untapped knowledge from data
- *Data Science* provides the tools, ML generates the models to create intelligent applications (or tasks)
Data Scientists

Train

Validate

Data

Machine Learning

Observe and Adjust

Configure

Train Model from Data

Model

Deploy Model For Intelligent Actions

New Data

Machine Learning App

Prediction / Inferences

Intelligent Action
Machine Learning Fundamentals
Machine Learning Concepts

Prediction Types

Training Modes

Models

Training and Testing
## Prediction Types

<table>
<thead>
<tr>
<th>Regression</th>
<th>Classification</th>
<th>Clustering</th>
<th>Recommender</th>
<th>Action</th>
</tr>
</thead>
</table>
| **Predict a Number**  
• House price | **Predict a Class**  
• Positive or negative sentiment  
• Intent  
• Spam or Ham? | **Group Related Items**  
• Related news articles | **Suggest Items**  
• Products on Amazon  
• Movies on Netflix | **Predict What To Do**  
• Game moves  
• Autonomous Vehicle driving |
## Learning Modes

<table>
<thead>
<tr>
<th>Supervised</th>
<th>Unsupervised</th>
<th>Reinforcement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Examples with labels or known answers provided</strong></td>
<td><strong>Examples with no labels or categories provided</strong></td>
<td><strong>Actions with feedback</strong></td>
</tr>
<tr>
<td>• Known house sales prices and house size, number of rooms, location, etc.</td>
<td>• Group examples based upon common features</td>
<td>• Positive or negative rewards for each action</td>
</tr>
<tr>
<td>• Customer reviews with known sentiment – positive or negative</td>
<td></td>
<td>• Tradeoff between Exploitation and Exploration</td>
</tr>
</tbody>
</table>

### OpenAI Gym BETA

A toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Go.
### Machine Learning Models – There are Many…

#### Decision Trees

1. **Income**
   - $>570k$
   - $<570k$

2. **Education**
   - $>16$
   - $<16$

3. **Married?**
   - Y
   - N

4. **Property Owner?**
   - Y
   - N

5. **Has Children?**
   - Y
   - N

- **Output Layer**

#### Linear Regression Models

- **Input Layer**
- **Hidden Layer-1**
- **Hidden Layer-2**

### Deep Neural Networks

- **Input Layer**
- **Hidden Layer-1**
- **Hidden Layer-2**
- **Output Layer**
Machine Learning Tribes

<table>
<thead>
<tr>
<th>Tribe</th>
<th>Origins</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbolists</td>
<td>Logic, Philosophy</td>
<td>Inverse Deduction</td>
</tr>
<tr>
<td>Connectionists (DL)</td>
<td>Neuroscience</td>
<td>Backpropogation</td>
</tr>
<tr>
<td>Evolutionaries</td>
<td>Evolutionary Biology</td>
<td>Genetic Programming</td>
</tr>
<tr>
<td>Bayesian</td>
<td>Statistics</td>
<td>Probabilistic Inference</td>
</tr>
<tr>
<td>Analogizers</td>
<td>Psychology</td>
<td>Kernel Machines</td>
</tr>
</tbody>
</table>
Example – Applying Machine Learning Concepts for Text Classification

- Prediction Types
- Training Modes
- Model Selection
- Model Training, Assessment, Optimization
**Machine Learning Process Example | Text Classification**

- **Collect Data**
- **Cleanse Data**
- **Define Features**
- **Train Classifier**
- **Assess Classifier**
- **Optimize Classifier**

**Data Extraction**
Extract documents

**Data Cleansing**
Validate documents are not empty, have a label and can be read.

**Data Preprocessing**
Utilization of different feature extraction techniques

**Classifier Training**
Train classifiers based gathered data

**Classifier Assessment**
Evaluate classification performance for different preprocessing and classification algorithms

**Classifier Optimization**
Optimize for automation or accuracy depending on client preferences
Collect Data

Collect Historical Data

Identify or Create 50+ Examples Per Class

<table>
<thead>
<tr>
<th>Training Example</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>I cannot login.</td>
<td>Password Reset</td>
</tr>
<tr>
<td>I’m locked out of the system.</td>
<td>Password Reset</td>
</tr>
<tr>
<td>I forgot my password.</td>
<td>Password Reset</td>
</tr>
<tr>
<td>I can’t access my documents.</td>
<td>User Rights</td>
</tr>
<tr>
<td>I can’t open my file.</td>
<td>User Rights</td>
</tr>
<tr>
<td>Permission is denied.</td>
<td>User Rights</td>
</tr>
<tr>
<td>The printer does not work.</td>
<td>Printer Malfunction</td>
</tr>
<tr>
<td>I can’t print.</td>
<td>Printer Malfunction</td>
</tr>
<tr>
<td>The printer is jammed.</td>
<td>Printer Malfunction</td>
</tr>
</tbody>
</table>
Cleanse Data

Remove Stopwords
Remove common words that add no value in understanding content.

a, an, and, as, at, from, is, it, are, that, there, to, were, will, with

Stem Text
Reduce inflectional forms and sometimes derivationally related forms of a word to their stem or common base form.

am, are, is → be
car, cars, car's, cars → car

Example: Email Data Cleansing
Remove Headers
Remove New Lines
Remove Confidentiality Footers
Remove Duplicates
Remove Empty Messages

to: helpdesk@accenture.com
Subject: FW: login problem

FYI

to: person@accenture.com
cc: otherppl@accenture.com
Subject: login problem

I was trying to log into my account and it’s not working. I don’t remember my master key for resetting so I can’t change this myself.

Cheers,
Mrs Smith

The contents of this email message and any attachments are intended solely for the addressee(s) and may contain confidential and/or privileged information and may be legally [...]

June 30, 2017 • Chicago Coder Conference
Copyright © 2017 Accenture All rights reserved.
Represent Data Numerically and Define Features

Documents to Vectors: Bag of Words
- Create a dictionary of all words present in corpus.
- Define a vector where each word is a dimension.
- Represent each document as a vector by counting the number of occurrences of each word.

Legend:
- **Doc 1**: I need to reset my password
- **Doc 2**: I want access to sharepoint

<table>
<thead>
<tr>
<th>Dictionary Word</th>
<th>Doc 1</th>
<th>Doc 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>need</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>reset</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>my</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>password</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>want</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>access</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sharepoint</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Better Representation: from Bag of Words to Term Frequency-Inverse Document Frequency (TF-IDF)

### Bag of Words to TF-IDF

- **Scale Down:** Globally Frequent Words
- **Scale Up:** Globally Rare Words

<table>
<thead>
<tr>
<th>Dictionary Word</th>
<th>Term Frequency (TF)</th>
<th>Inverse Document Frequency (IDF)</th>
<th>TF-IDF = TF x IDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>serendipity</td>
<td>5</td>
<td>6</td>
<td>30</td>
</tr>
</tbody>
</table>

Number of Documents: **1,000**

**Example 1**
Globally Common Word: *the*
Num Docs containing *the*: **1000**

\[
IDF = \log\left(\frac{1000}{1 + 1000}\right) \approx 0
\]

**Example 2**
Globally Uncommon Word: *serendipity*
Num Docs containing *serendipity*: **2**

\[
IDF = \log\left(\frac{1000}{1 + 2}\right) \approx 6
\]
Train Classifier: First, Select Model

- Datacamp Scikit-Learn Cheat Sheet
  - [www.datacamp.com/community/blog/scikit-learn-cheat-sheet](http://www.datacamp.com/community/blog/scikit-learn-cheat-sheet)
- Scikit-Learn Tutorial Cheat Sheet
Different Models Perform Better on Different Data Sets

Naïve Bayes - generative

Support Vector Machine – discriminative
Train Classifier

**Training Examples**

<table>
<thead>
<tr>
<th>Text Example</th>
<th>True Intent</th>
</tr>
</thead>
<tbody>
<tr>
<td>I've moved</td>
<td>Change Address</td>
</tr>
<tr>
<td>I have a new address</td>
<td>Change Address</td>
</tr>
<tr>
<td>I would like to change my address</td>
<td>Change Address</td>
</tr>
<tr>
<td>Can I upgrade my phone?</td>
<td>Phone Upgrade</td>
</tr>
<tr>
<td>Am I eligible for an upgrade</td>
<td>Phone Upgrade</td>
</tr>
<tr>
<td>I need a new phone</td>
<td>Phone Upgrade</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Representation: Vector of Word Counts**

<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>Would</td>
<td>1</td>
</tr>
<tr>
<td>Like</td>
<td>1</td>
</tr>
<tr>
<td>To</td>
<td>1</td>
</tr>
<tr>
<td>Change</td>
<td>1</td>
</tr>
<tr>
<td>My</td>
<td>1</td>
</tr>
<tr>
<td>Address</td>
<td>1</td>
</tr>
<tr>
<td>Upgrade</td>
<td>0</td>
</tr>
<tr>
<td>Eligible</td>
<td>0</td>
</tr>
<tr>
<td>New</td>
<td>0</td>
</tr>
<tr>
<td>Phone</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Machine Learning Model**

**Predicted Intent**

**Loss**

**Learning Algorithm Updates Model to Minimize Loss**

**Correct Intent**
Support vector machine classification: Two classes of points with 3 candidate linear separators.
Support vector machine classification: The maximum margin separator is at the midpoint of the margin. The support vector (large circles) are the samples closest to the separator.
Support vector machine classification: A two-dimensional training set with a circular margin separating the two feature values. The circular decision boundary in 2-d becomes a linear decision boundary in 3-d (see next).
Support vector machine classification: The margin in the previous slide can be represented as a plane in three dimensions discriminating the feature values. Values above the plane are one set; values below the plane are in the other.
Assess Classifier Performance

Confusion Matrix
Displays performance per label and how incorrect (confused) labels have been assigned.

<table>
<thead>
<tr>
<th>True Class</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>TP</td>
</tr>
<tr>
<td>B</td>
<td>FP</td>
</tr>
<tr>
<td>Precision</td>
<td>TP/TP+FP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TP+TN All</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>TP</td>
</tr>
<tr>
<td>B</td>
<td>FP</td>
</tr>
</tbody>
</table>

Accuracy

<table>
<thead>
<tr>
<th>True Class</th>
<th>Predicted Class</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Precision</th>
<th>79% 72% 100%</th>
</tr>
</thead>
</table>
Assess Classifier Performance
Tradeoff between Precision and Recall

Which oval best picks out blue balls?

Precision
Fraction of positive predictions that are correct

Recall
Fraction of positive data correctly identified by the model
Where to learn more

- **Data Science**: William Chen compiled a list of free books
- **Machine Learning Wiki Page** (links to online courses, books, data sets, etc. for ML)
  - [https://www.reddit.com/r/MachineLearning/wiki/index](https://www.reddit.com/r/MachineLearning/wiki/index)
- **Joseph Misiti’s Curated List** of Machine Learning frameworks, libraries, etc.
  - [https://github.com/josephmisiti/awesome-machine-learning](https://github.com/josephmisiti/awesome-machine-learning)
- **Python Unit testing**
  - [http://www.diveintopython3.net/unit-testing.html](http://www.diveintopython3.net/unit-testing.html)
- **Pandas data frame tutorials**
- **scikit-learn**
- **Matplotlib for plotting in Python**
  - [http://matplotlib.org/resources/index.html](http://matplotlib.org/resources/index.html)
THANK YOU!

John Shiner
Introduction to Machine Learning
Track #Other • 3:45 PM - 4:45 PM pm • Room #200

Social
/in/johnshiner/
@JohnShiner2016

Email
john.shiner@accenture.com