Disclaimer

These opinions and thoughts are my own, and may or may not reflect the opinions of the company that I work for.

Python Data Analytics

andrew.zhang@ibm.com
Data and AI, IBM Cognitive Systems

RBS – October 26, 2019

Agenda

- **Introduction**
- **Data Analysis with Pandas**
- **Data Visualization with Matplotlib**
- **Machine Learning with Scikit-learn**
- **Bonus: Power AI Vision**

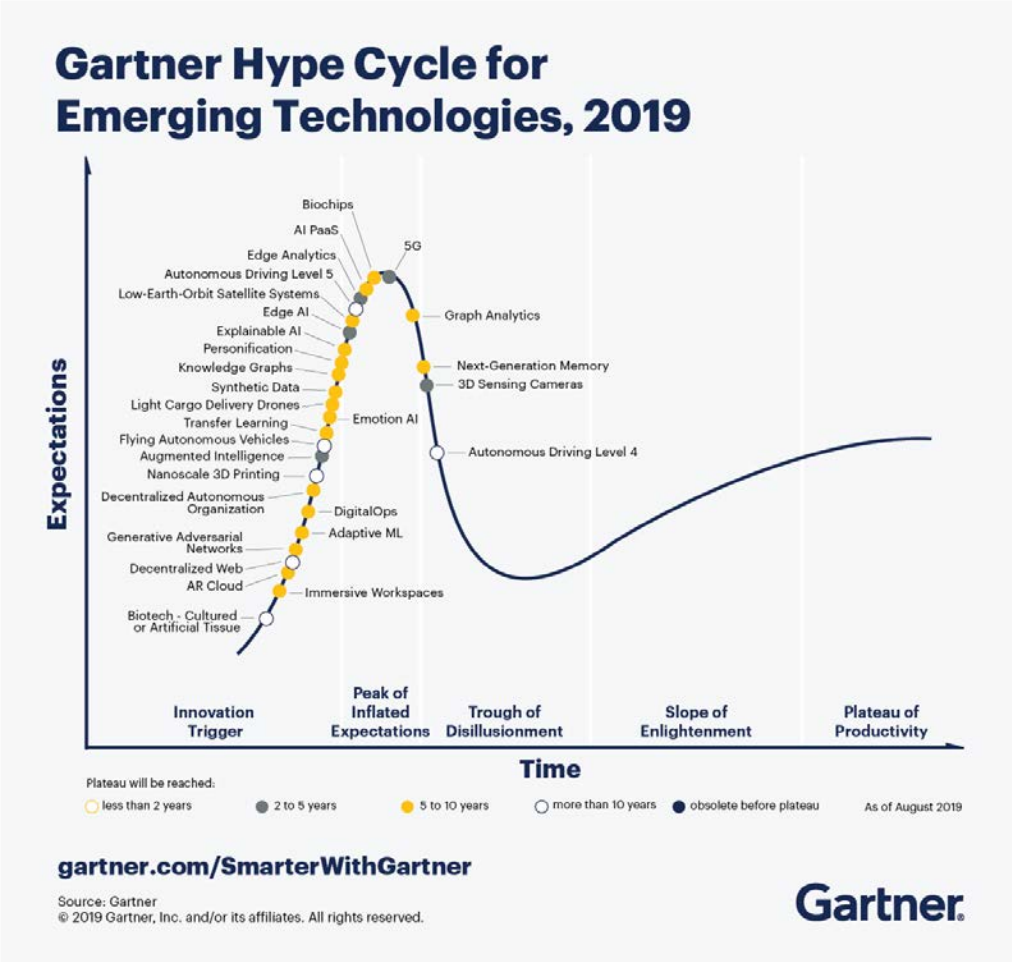
About me

- Open Source
- Big Data Analytics
- Data and AI
- High Performance Computing

"We have lots of information technology. We just don't have any information."

Part One - Introduction

Future of Technologies



1. Cloud
2. Data
3. Artificial Intelligence

Why are enterprises struggling to capture the value of Data and AI?

Data

- Data resides in silos & difficult to access
- Unstructured and external data wasn't considered

Governance

- If the data isn't secure, self-service isn't a reality
- Challenge understanding data lineage and getting to a system of truth

Skills

- Data Science skills are in low supply and high demand
- Nurturing new data professionals is challenging

Tools & Infrastructure

- Need an environment that enables a "fail fast" approach
- Discrete tools present barriers to productivity

Data Analytics Magic Quadrant 2018

Part Two
Python Data Analytics

What is the most popular programming language nowadays?

Introducing Python

Play 4mins Video

Introducing Python

*"Python is **powerful**... and fast**;** plays well with others; runs everywhere; is **friendly** & easy to learn; is **Open**."*

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

matplotlib

Version 3.1.1

scikit learn

IP[y]:

jupyter

https://www.python.org/

Data Analysis

Play 2 mins Video

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

- A fast and efficient **DataFrame** object for data manipulation
- **Reading and writing data** : CSV and text files, Microsoft Excel, SQL databases, and the fast HDF5 format
- **Data alignment** and handling of **missing data**
- **Reshaping** and pivoting of data sets
- **Slicing**, **indexing**, and **subsetting** of large data sets
- **Group by**, **merging and joining** of data sets;
- Python with *pandas* is in use in a wide variety of **academic and commercial** domains, including Finance, Neuroscience, Economics, Statistics, Advertising, Web Analytics, and more.

https://pandas.pydata.org/

Data Visualization

matplotlib

Version 3.1.1

- Python 2D plotting library which produces **publication quality** figures
- **Interactive environments** with Python shell, IPython, Jupyter notebook, and web application servers
- Generate plots, histograms, bar charts, scatter plots, etc., with just **a few lines of code**
- **Simple plotting** pyplot module provides a MATLAB-like interface
- **Full control** of line styles, font properties, axes properties

https://matplotlib.org/

Machine Learning

Play Video

- Most popular machine learning library in Python
- Built on NumPy, SciPy, and matplotlib
- **Classification:** Identifying to which **category** an object belongs to such as spam detection, image recognition
- **Regression:** Predicting a **continuous-valued attribute** associated with an object such as energy consupmption, stock price
- **Clustering:** Automatic **grouping** of similar objects into sets such as customer segmentation and grouping experiment outcome
- **Dimensionality reduction:** Reducing the number of random variables to consider such as visualization, increased efficiency

https://scikit-learn.org/stable/

Lab Setup

Download & Install Anaconda

https://www.anaconda.com/distribution/

Jupyter Notebook

In [1]:
```
import this
```

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

jupyter RBS-Workshop-Lab 1- Car Price (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Markdown

Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Introduction

Data Acquisition

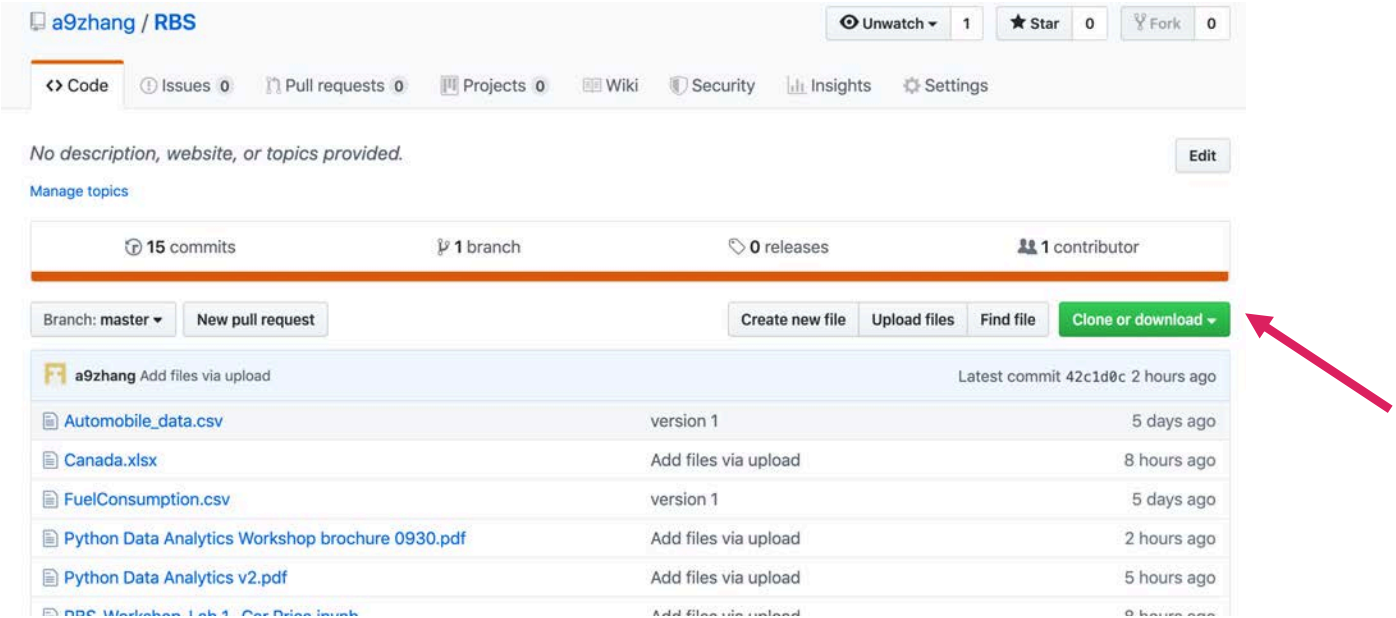
In [2]:
```
path = 'https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data
df_1 = pd.read_csv(path)
df_1.head()
```

Out[2]:

	symboling	normalized-losses	make	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	leng

Download and Import Notebooks

https://github.com/a9zhang/RBS



Lab Exercises

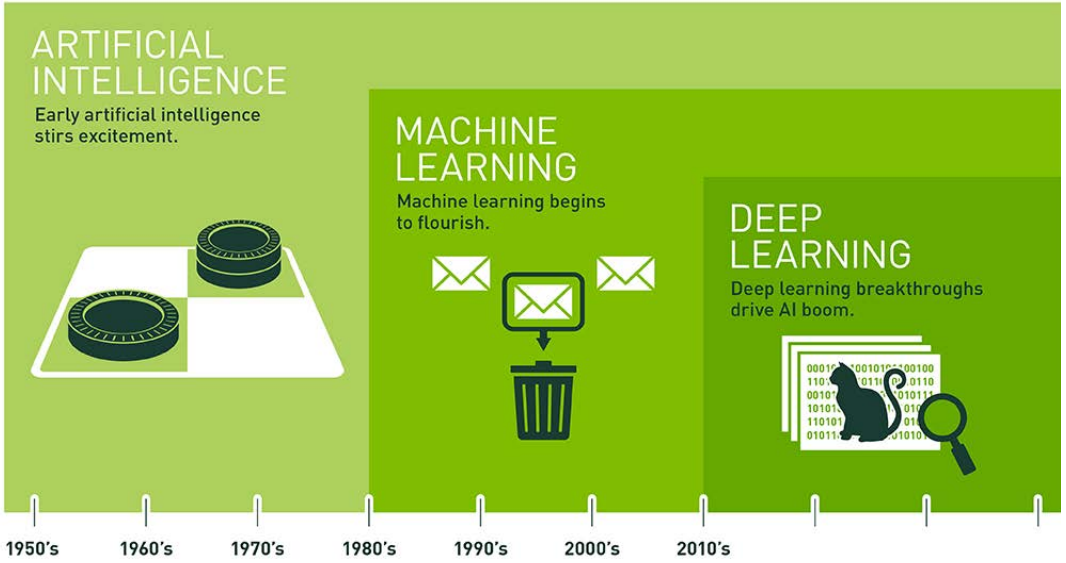
Lab Exercises (3 hours)

- **Lab 1: Data Analysis with Pandas**
- **Lunch**
- **Lab 2: Data Visualization with Matplotlib**
- **Lab3: Machine Learning with Scikit-learn**

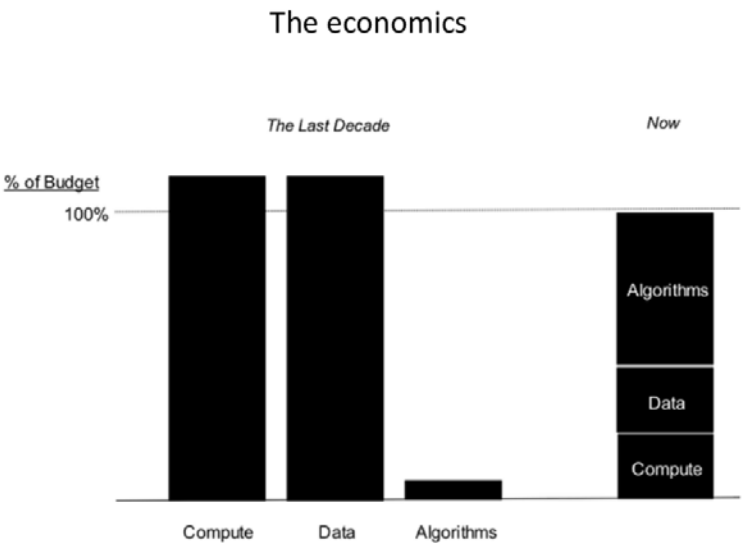
Bonus

https://www.youtube.com/watch?v=CkVZRMG6pc4&feature=youtu.be

Why Data and AI, Why Now?



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.



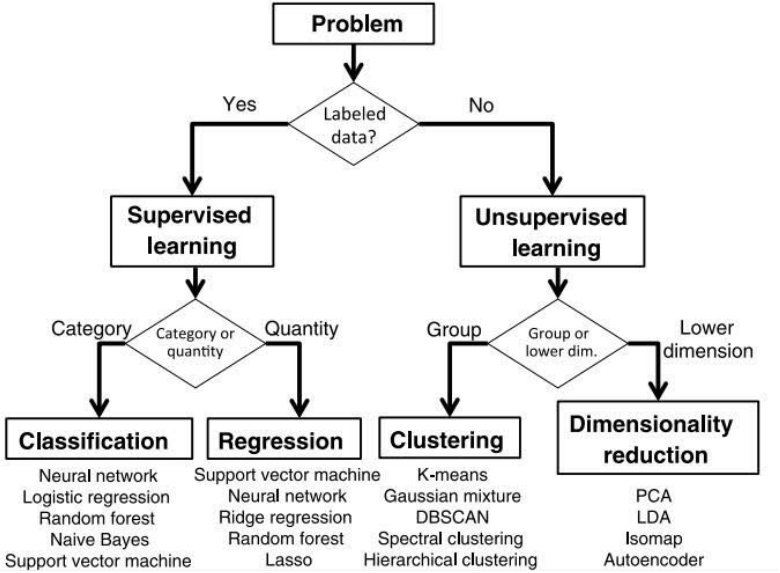
Machine Learning vs Deep Learning

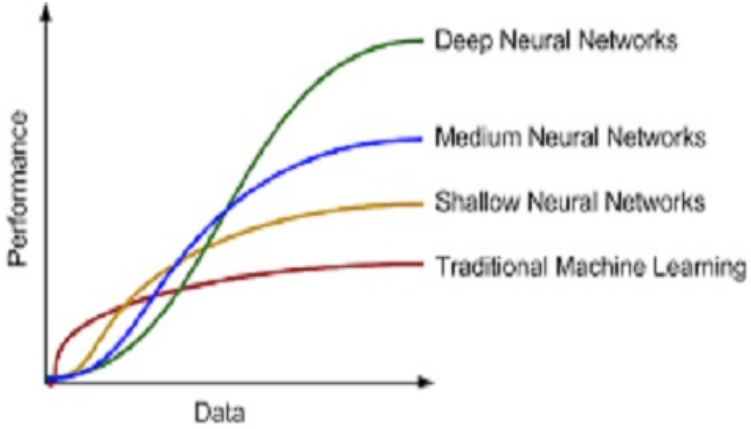
Machine Learning

- Traditional ML requires manual feature extraction/engineering
- Feature extraction for unstructured data is very difficult

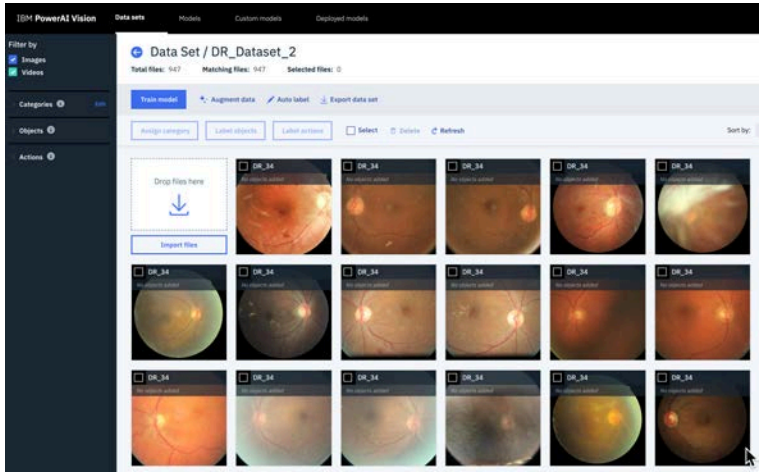
Deep Learning

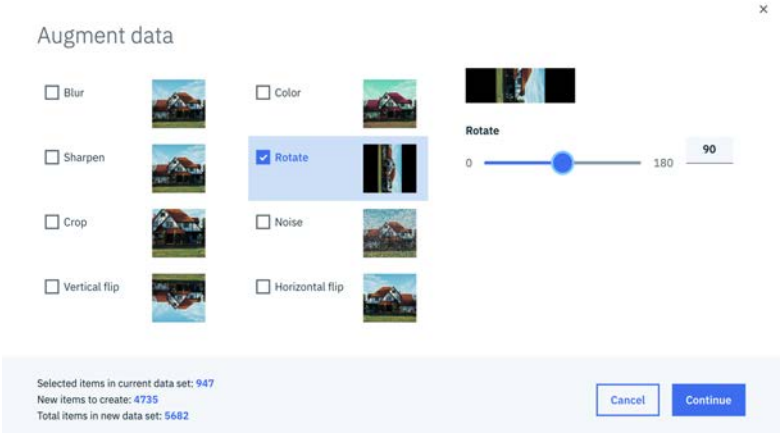
- Deep learning can automatically learn features in data
- Deep learning is largely a "black box" technique, updating learned weights at each layer





Diabetic Retinopathy Detection using IBM Power AI Vision





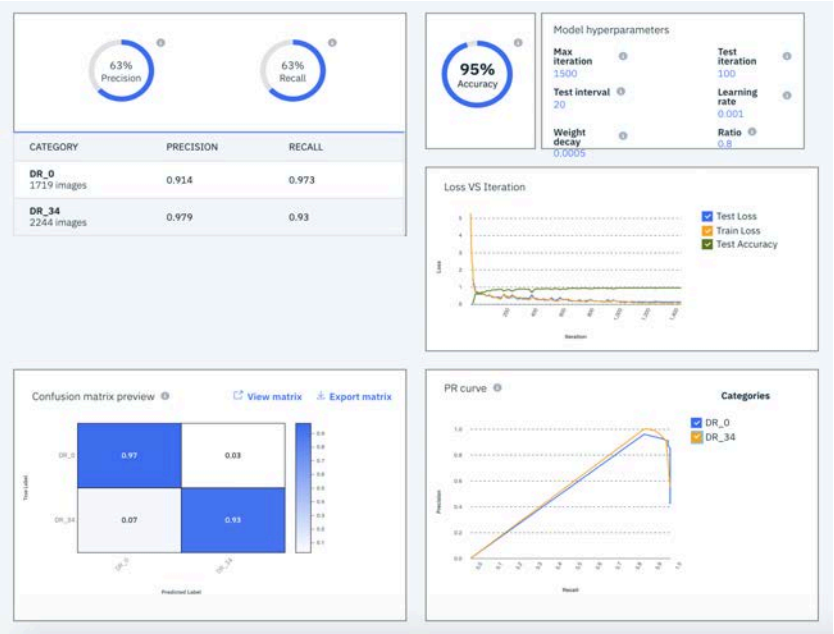
35,000 + images of various classes

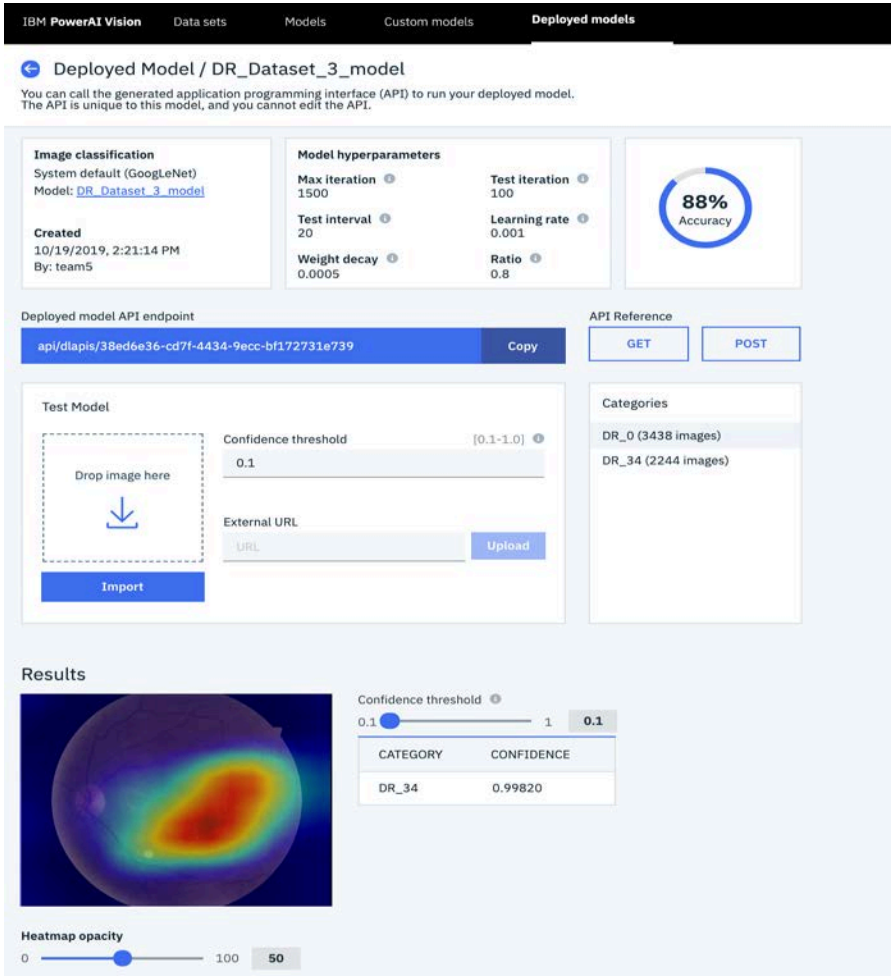
0 - No DR
1 - Mild
2 - Moderate
3 - Severe
4 - Proliferative DR

Data Augmentation – create more data

Resize, Crop, Rotate, Flip, Translation…

Diabetic Retinopathy Detection using Power AI Vision

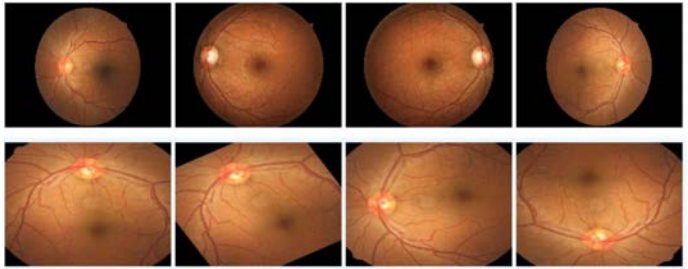




Train machine learning model with high accuracy **without coding**

Detect new images – DR vs No_DR images with a trained model

Diabetic Retinopathy Detection using Python



Python and Open Source

- **Exploratory Data Analysis**
- **Crop and Resize Images on cloud**
- **Rotate and Mirror Images**
- **Neural Network Architecture**

Metric	Value
Accuracy (Train)	82%
Accuracy (Test)	80%
Precision	88%
Recall	77%

Takeaways

Play Video

Bar Chart Racing in Python ~In roughly less than 50 lines of code

Questions ?

Backup Slides