Python Data Analytics

Andrew Zhang
Open Source Analytics, IBM
2019/10/26, RBS

Agenda

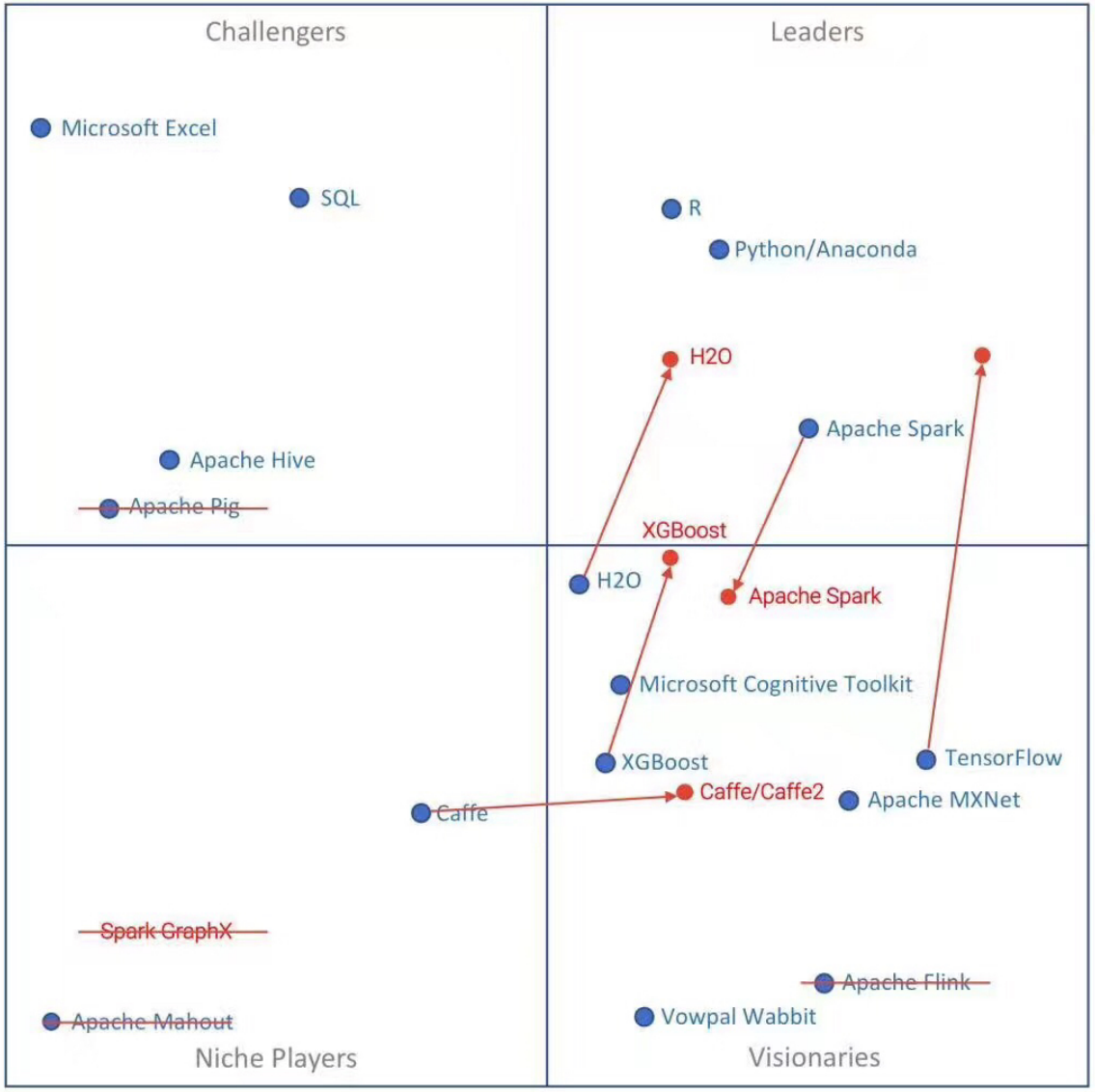
- **Introduction**
- **Data Analysis with Pandas**
- **Data Visualization with Matplotlib**
- **Machine Learning with Scikit-learn**
- **Bonus: Power AI Vision**

About me

- Open Source
- Big Data Analytics
- Data Science and Machine Learning
- High Performance Computing

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

Data Analytics Magic Quadrant (2018)



What is the most popular programming language nowadays?

Introducing Python

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

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

- 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

- 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!
```

Download and Import Notebooks

https://github.com/a9zhang/RBS

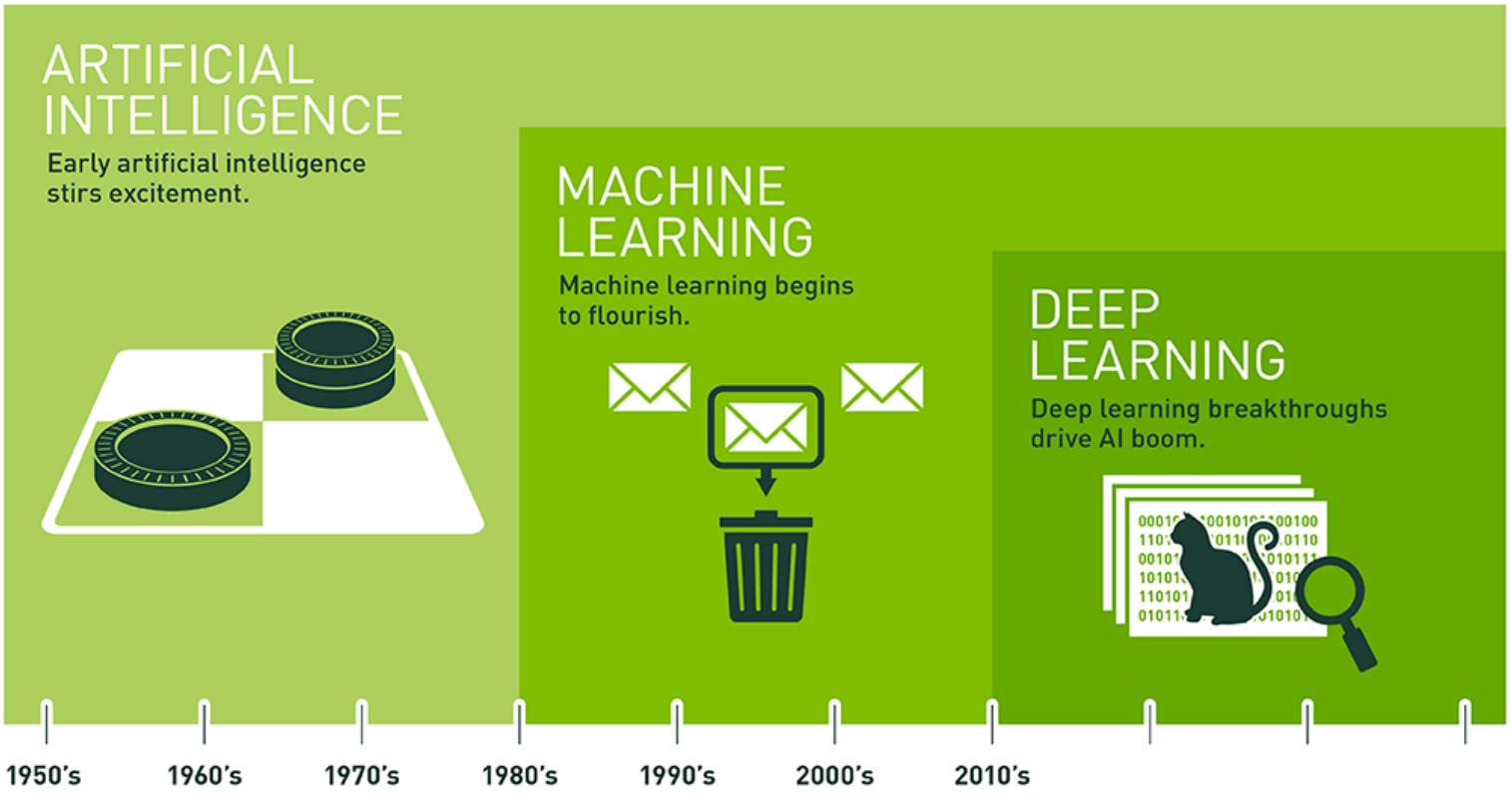
Lab Exercises

Labs (3 hours)

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

Bonus

Machine Learning, Deep Learning and AI



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.

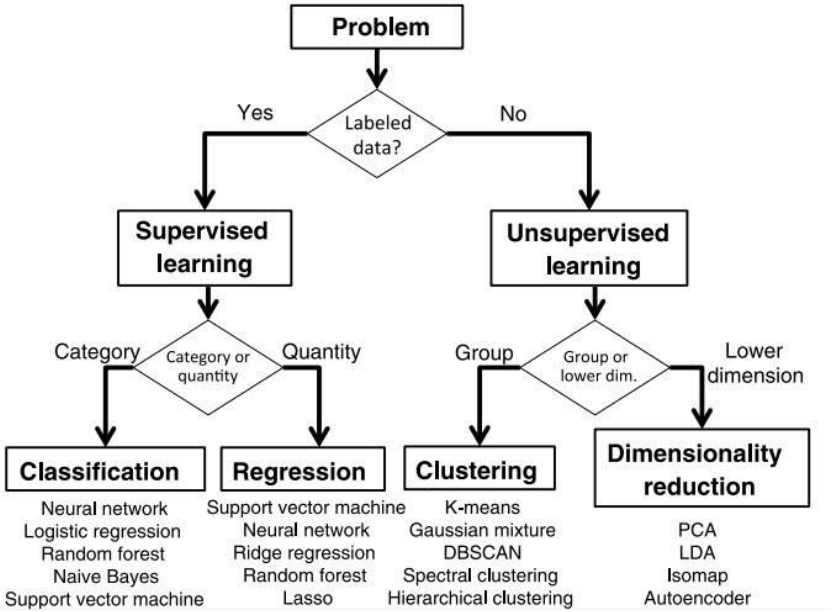
Machine Learning vs Deep Learning

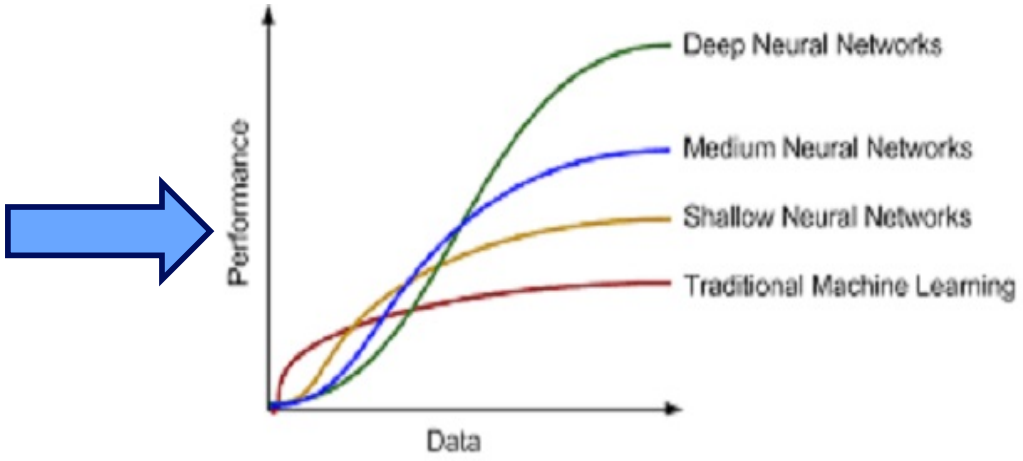
Machine Learning

- 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

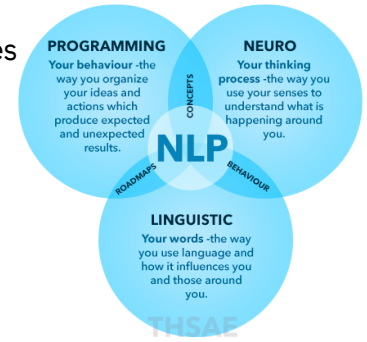




Popular Machine Learning Services

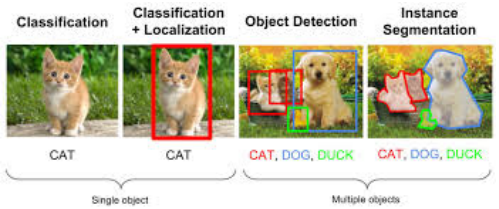
NLP

- Natural Language Processing Services
- Entity Extraction
- Key Phrase Extraction
- Sentiment Analysis
- Syntax Analysis
- Topic Modeling
- Multiple Language Support
- Parts of Speech



Visual Recognition

- Object Detection
- Scene Detection
- Facial Recognition
- Flag In-appropiate Content
- Facial Analysis
- Celebrity Recognition
- Logo Detection
- Text Recognition
- Web Detection
- Landmark Detection
- Dominant Color Detection
- Thumbnail Generation

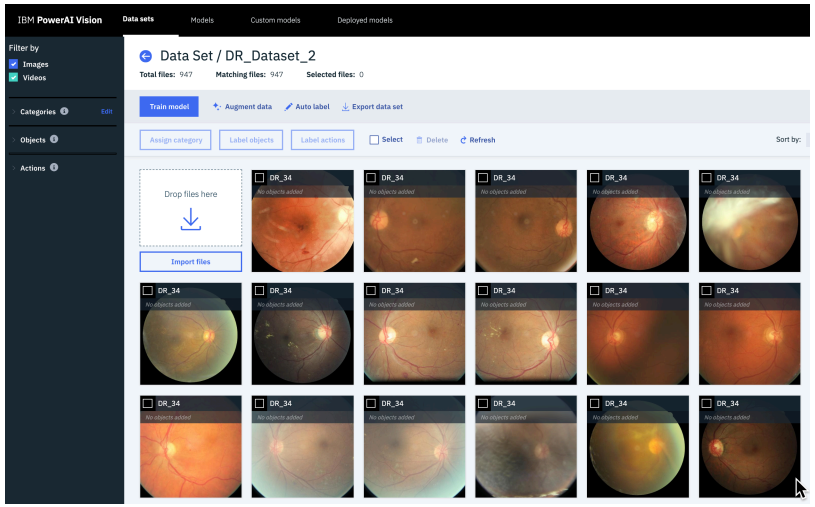


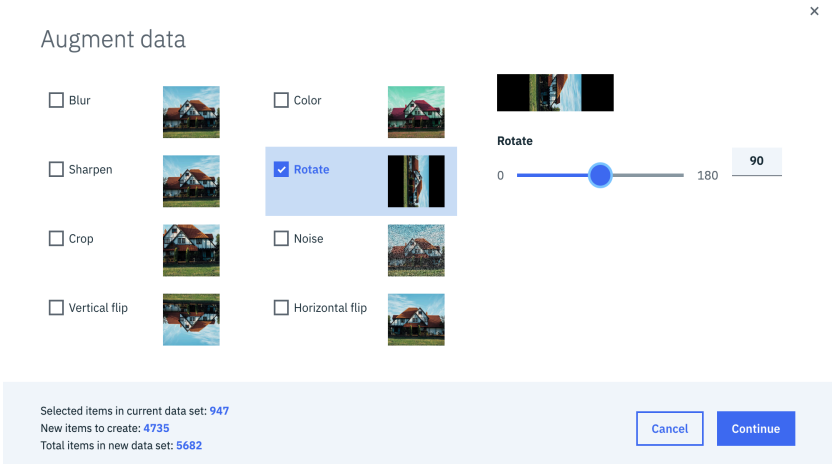
Speech

- SSML
- Multiple Language
- Format
- Automatic Speech Recognition (ASR)
- Noisy Accuracy

[Machine Learning (ML) Services from Various Cloud ML Service Providers]

Diabetic Retinopathy Detection using Power AI Vision





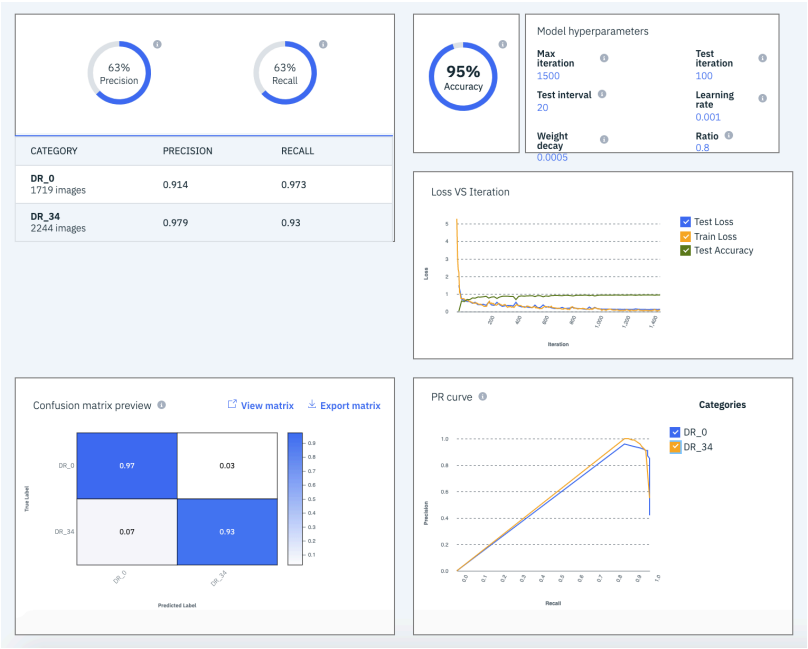
35,000 + images of various classes

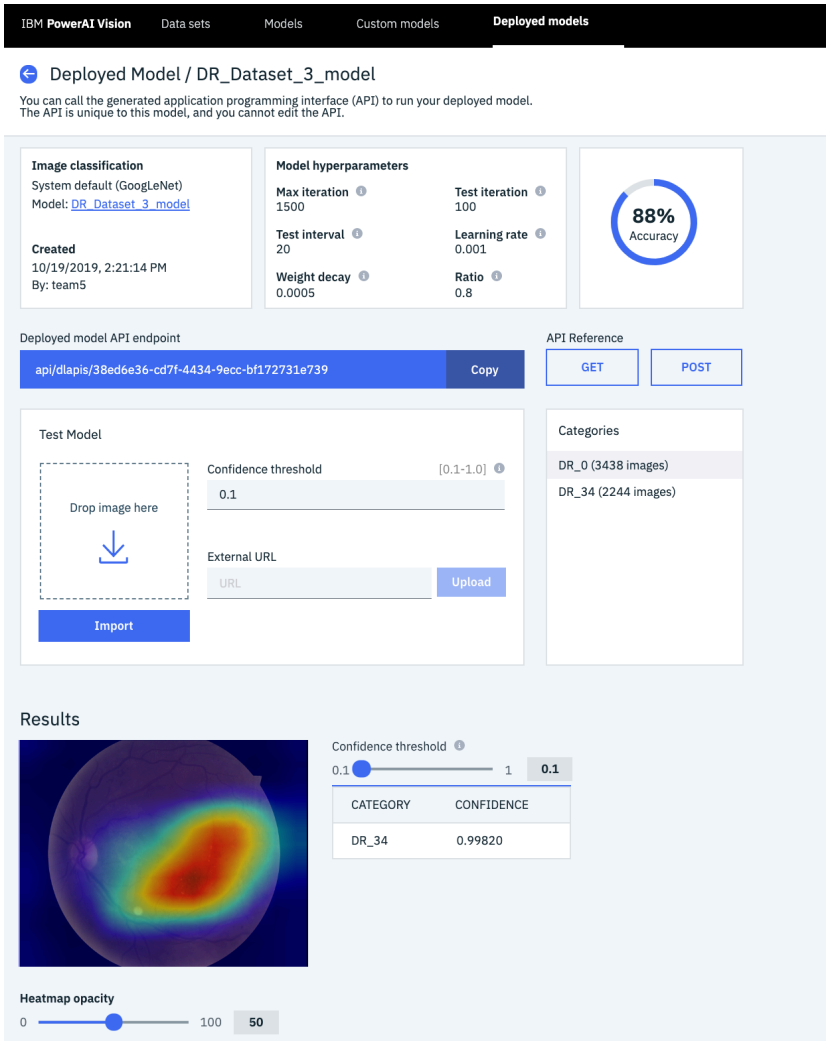
0 - No DR
1 - Mild
2 - Moderate
3 - Severe
4 - Proliferative DR

Data Augmentation – create more data

Resize, Crop, Rotate, Flip, Translation...

Diabetic Retinopathy Detection using Power AI Vision

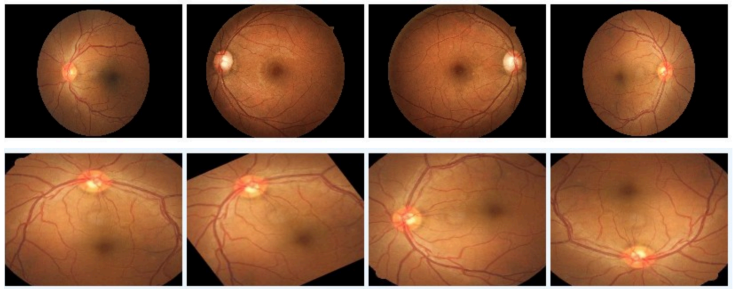




Train machine learning model with high accuracy without coding

Detect new images – DR vs No_DR 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%

Backup Slides

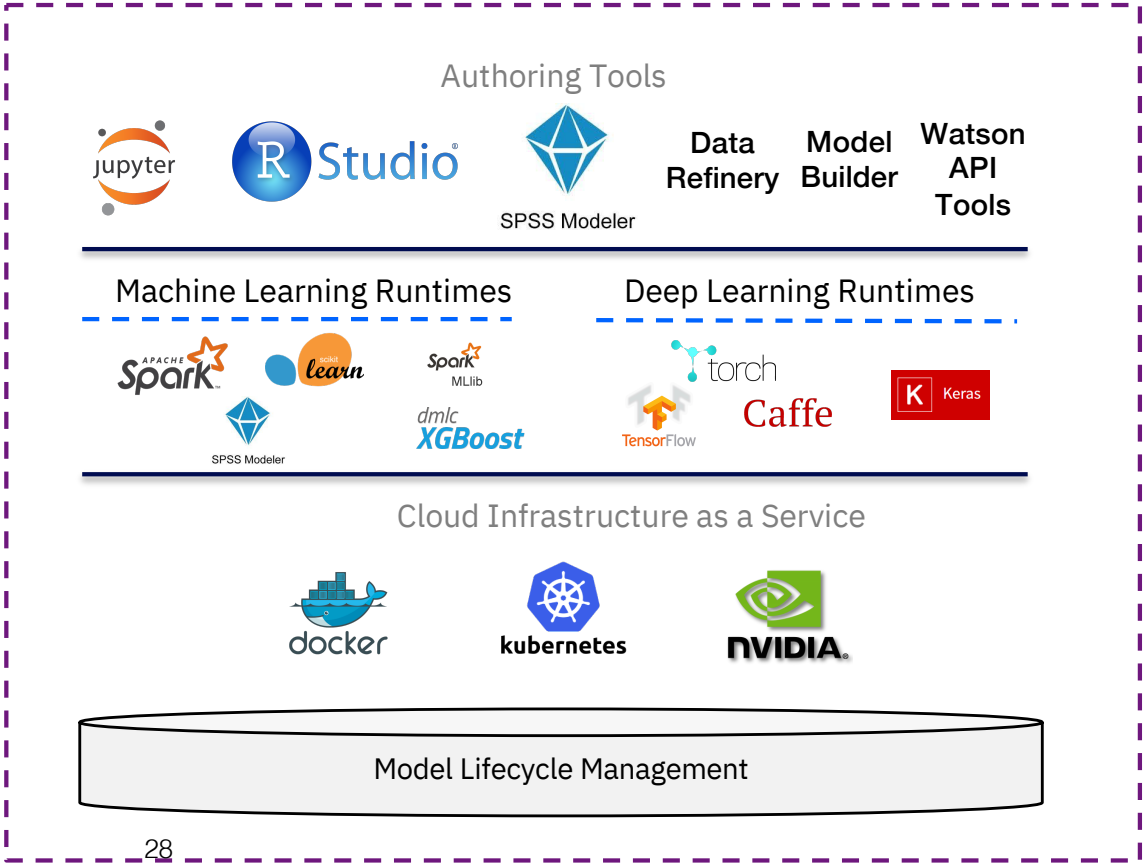
Watson Studio

Comprehensive set of tools for the end-to-end AI workflow

- Create, collaborate, deploy, and monitor
- Best of breed open source & IBM tools
- Code (R, Python or Scala) and no-code/visual modeling tools

- Most popular open source frameworks
- IBM best-in-class frameworks

- Fully managed service
- Container-based resource management
- Elastic pay as you go CPU/GPU power



Machine Learning, Deep Learning and AI

The hardest part of AI isn't AI

"Hidden Technical Debt in Machine Learning Systems ", Google NIPS

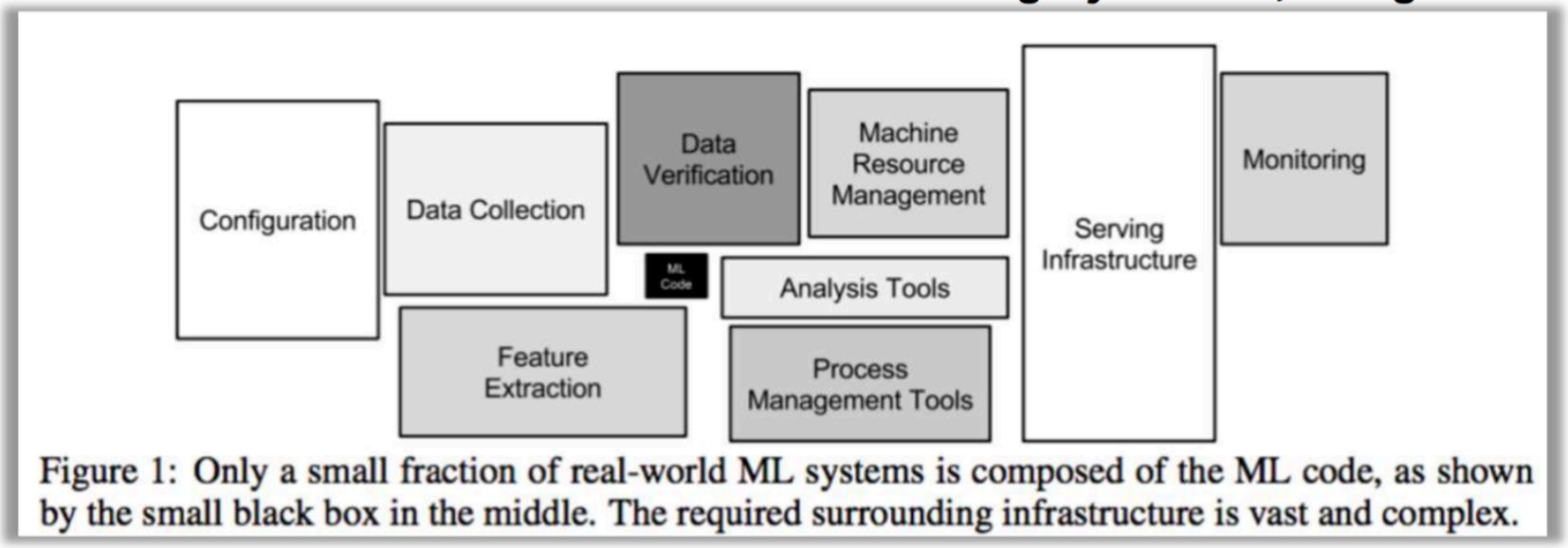


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Bonus

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

Pandas Introduction
Question #1: check the bottom 10 rows of data frame "df".
Question #2: Find the name of the columns of the dataframe
Question #3: You can select the columns of a data frame by indicating the name of each column, for example, you can select the three columns as follows:
dataframe[[' column 1 ',column 2', 'column 3']]
Where "column" is the name of the column, you can apply the method ".describe()" to get the statistics of those columns as follows:
dataframe[[' column 1 ',column 2', 'column 3']].describe()
Apply the method to ".describe()" to the columns 'length' and 'compression-ratio'.

Data Wrangling
Question #1: According to the example above, replace NaN in "stroke" column by mean.
Question #2: According to the example above, transform mpg to L/100km in the column of "highway-mpg", and change the name of column to "highway-L/100km".
Question #3: According to the example above, normalize the column "height".
Question #4: As above, create indicator variable to the column of "aspiration": "std" to 0, while "turbo" to 1.
Question #5: Merge the new dataframe to the original dataframe then drop the column 'aspiration'

Exploratory Data Analysis
Question #1: What is the data type of the column "peak-rpm"?
Question #2: Find the correlation between the following columns: bore, stroke,compression-ratio , and horsepower.
Hint: if you would like to select those columns use the following syntax: df[['bore','stroke' ,'compression-ratio','horsepower']]
Question 3 a): Find the correlation between x="stroke", y="price".
Hint: if you would like to select those columns use the following syntax: df[["stroke","price"]]
Question 3 b): Given the correlation results between "price" and "stroke" do you expect a linear relationship?
Verify your results using the function "regplot()".
Question 4: Use the "groupby" function to find the average "price" of each car based on "body-style" ?

Matplotlib Introduction
Question: Plot a line graph of immigration from Haiti using df.plot()
Question: Let's compare the number of immigrants from India and China from 1980 to 2013.
Question: Compare the trend of top 5 countries that contributed the most to immigration to Canada.

Area Plots, Histograms, and Bar Plots
Question: Use the scripting layer to create a stacked area plot of the 5 countries that contributed the least to immigration to Canada **from** 1980 to 2013. Use a transparency value of 0.45.
Question: Use the artist layer to create an unstacked area plot of the 5 countries that contributed the least to immigration to Canada **from** 1980 to 2013. Use a transparency value of 0.55.
Question: What is the frequency distribution of the number (population) of new immigrants from the various countries to Canada in 2013?
Question: What is the immigration distribution for Denmark, Norway, and Sweden for years 1980 - 2013?
Question: Use the scripting layer to display the immigration distribution for Greece, Albania, and Bulgaria for years 1980 - 2013? Use an overlapping plot with 15 bins and a transparency value of 0.35.
Question: Let's compare the number of Icelandic immigrants (country = 'Iceland') to Canada from year 1980 to 2013.
Question: Using the scripting layter and the df_can dataset, create a *horizontal* bar plot showing the *total* number of immigrants to Canada from the top 15 countries, for the period 1980 - 2013. Label each country with the total immigrant count.

Pie Charts, Box Plots, Scatter Plots, and Bubble Plots
Question: Using a pie chart, explore the proportion (percentage) of new immigrants grouped by continents in the year 2013.
Question: Compare the distribution of the number of new immigrants from India and China for the period 1980 - 2013.
Question: Create a box plot to visualize the distribution of the top 15 countries (based on total immigration) grouped by the *decades* 1980s, 1990s, and 2000s.
Question: Create a scatter plot of the total immigration from Denmark, Norway, and Sweden to Canada from 1980 to 2013?
Question: Previously in this lab, we created box plots to compare immigration from China and India to Canada. Create bubble plots of immigration from China and India to visualize any differences with time from 1980 to 2013. You can use **df_can_t** that we defined and used in the previous example.

ML0101EN-Reg-Simple-Linear-Regression-Co2-py-v1

Practice

plot CYLINDER vs the Emission, to see how linear is their relation:

ML0101EN-1-Reg-Mulitple-Linear-Regression-Co2-py-v1

Practice

Try to use a multiple linear regression with the same dataset but this time use __FUEL CONSUMPTION in CITY__ and __FUEL CONSUMPTION in HWY__ instead of FUELCONSUMPTION_COMB. Does it result in better accuracy?

ML0101EN-2-Clas-Decision-Trees-drug-py-v1

Practice

What is the size of data?

Practice

Print the shape of X_trainset and y_trainset. Ensure that the dimensions match

In [10]:

Practice

Can you calculate the accuracy score without sklearn ?

ML0101EN-3-Clus-K-Means-Customer-Seg-py-v1

Practice

Try to cluster the above dataset into 3 clusters.

Notice: do not generate data again, use the same dataset as above.

ML0101EN-4-Clas-Logistic-Reg-churn-py-v1

\Practice

How many rows and columns are in this dataset in total? What are the name of columns?

Practice

Try to build Logistic Regression model again for the same dataset, but this time, use different __solver__ and __regularization__ values? What is new __logLoss__ value?

ML0101EN-5-RecSys-Content-Based-movies-py-v1