Data Science with python

MNIT, Jaipur

Falak Shah
Lead Research Scientist
Infocusp Innovations Pvt. Ltd.

About me

I’m passionate about application oriented research (and meditation :). I have 5 years of experience in using Machine learning for a wide array of applications.

I grew up in Ahmedabad, am an avid reader, and love travelling to mountains.

Not a speaker - please speak up!

Your background in python/ ML/ data science?

About IIPL

- Financial modelling
- Sifting through legal documents
- Biomedical applications - mix of DSP/ ML
- Code translation using DL
- Research (most exciting!) - Music generation, Visualizing CNNs, Kaggle DS for good, LSTMs (writing like Shakespeare), optimizers and many more

What you could get out of my sessions

- Data Science fundamentals
- Kaggle contests starter tips
- Using python for ML
- Using python (TF/ Keras) for Deep Learning
- Using/ contributing to python / open source software
- Random tips/ questions we've asked for ML/ Data scientist position

Motivation: Examples of data science applications

PASSNYC Data Science for good challenge

https://www.kaggle.com/infocusp/holographic-view-of-underperforming-schools

https://www.kaggle.com/infocusp/recommendations-to-passnyc-based-on-data-analysis

Purely data science - no ML

Careervillage: Mapping professionals to likely questions

---

https://www.kaggle.com/infocusp/deepdive-into-careervillage/data

Not just about coding either: https://www.kaggle.com/danielbecker/careervillage-org-recommendation-engine

But once you have the idea clear, python to the rescue

Contents

Loading data - knowing the various data formats

Organizing/ Cleaning data

Analysing data

Summarizing the stats

Visualizing

Open source contributions

Disclaimer: Can’t show everything in detail in 3 hours so - pointers to what you should know and where you can learn it from. But we’ll do pandas hands on.

Loading data

Most commonly used formats: csv, Hierarchical Data Format (HDF5), sql, json, images, html, mp3, mp4

Pandas has functions to read most tabular formats directly into dataframes

Possible variations : Multi index, data too large to fit into memory - use spark (beyond scope of this tutorial)

Temporary solution : Read it in chunk wise and write out to separate files

```
pd.read_csv
pd.read_excel
pd.read_json
pd.read_hdf

import zipfile
archive = zipfile.ZipFile('T.zip', 'r')
df = archive.read('train.csv')

XML:
import xml.etree.ElementTree as ET
tree = ET.parse('train.xml')

HTMLs: Use beautifulsoup

Images:
Import cv2
cv2.imread(‘image.png’)
```

Organizing/ Cleaning the data

- Connect related columns from different files
- Fill NaNs
- Remove outliers
- Fix noisy samples (incorrect data type)
- Get Rid of Extra Spaces
- Fill in blank data (mean/ median/ zeros)
- Convert numbers stored as text to floats/ int
- Categorical data to numbers

Quality data beats fancy algorithms

https://colab.research.google.com/drive/1wvWbtjcxIqmxFcMgJFxehZlXAg7NrWQb

Part 2: Visualization and open source

Visualizing

Horde of options:

Matplotlib

Seaborn

Plotly

QT (for real time), Tensorboard

Bokeh

And many more - learn from kaggle and debug using SO

Examples

Kaggle notebooks we saw in the previous session

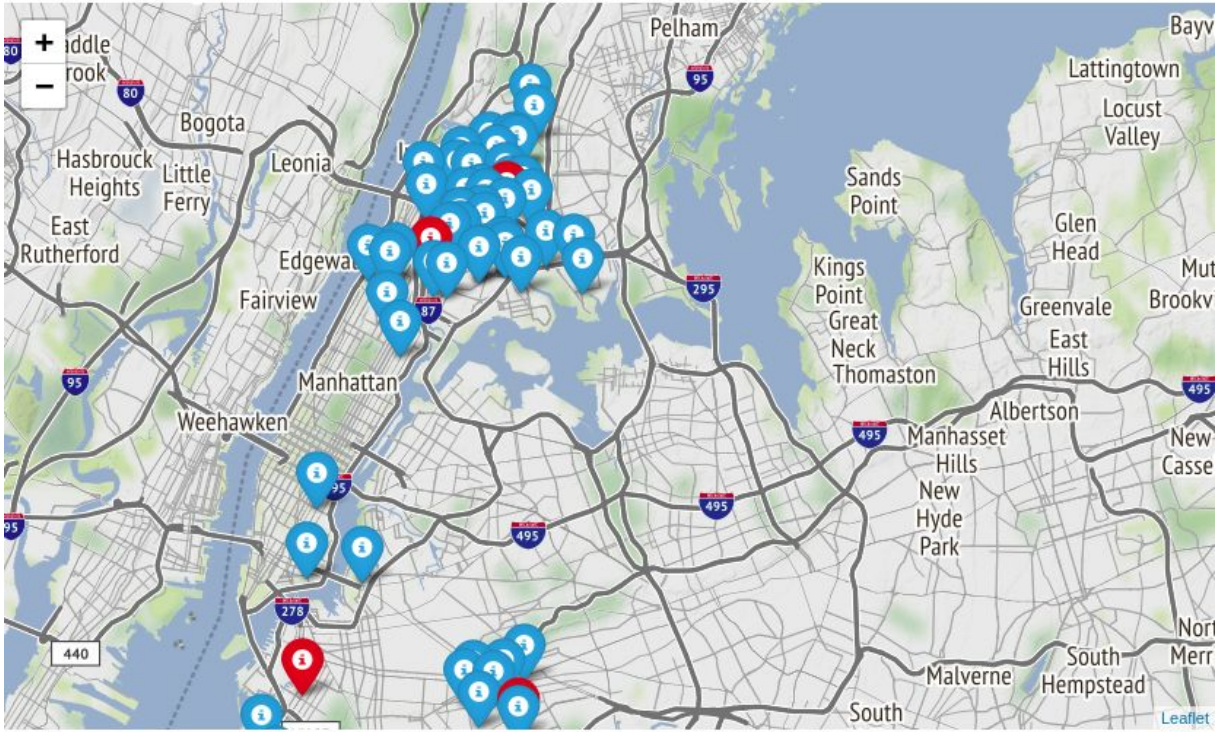
https://github.com/InFoCusp/lstm_deepdive

https://github.com/InFoCusp/tf_cnnvis (Just images)

Gradient descent based optimizers to visualize loss

TSNE: Link

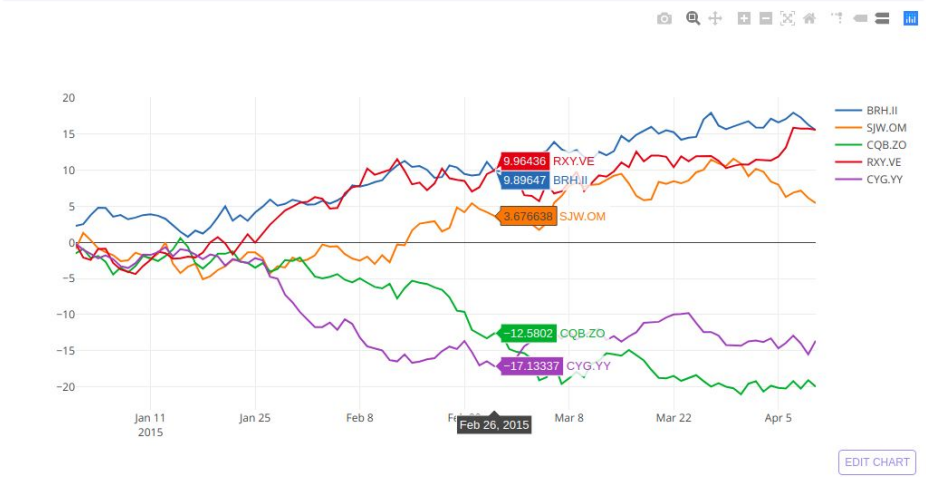
Not just plots either

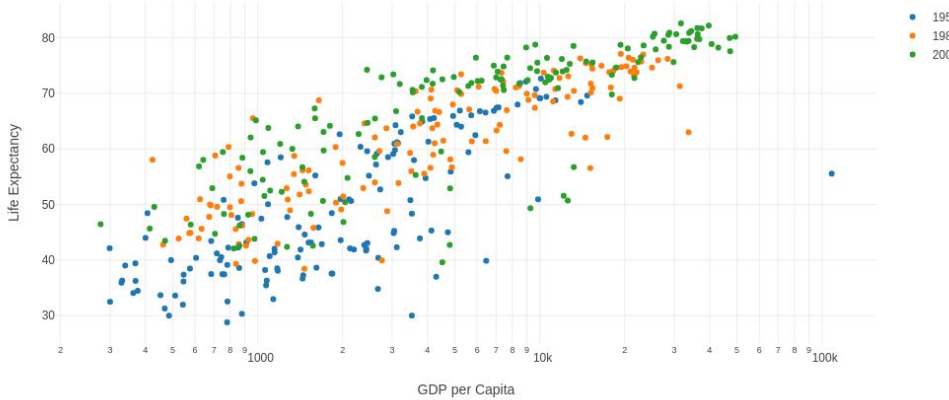


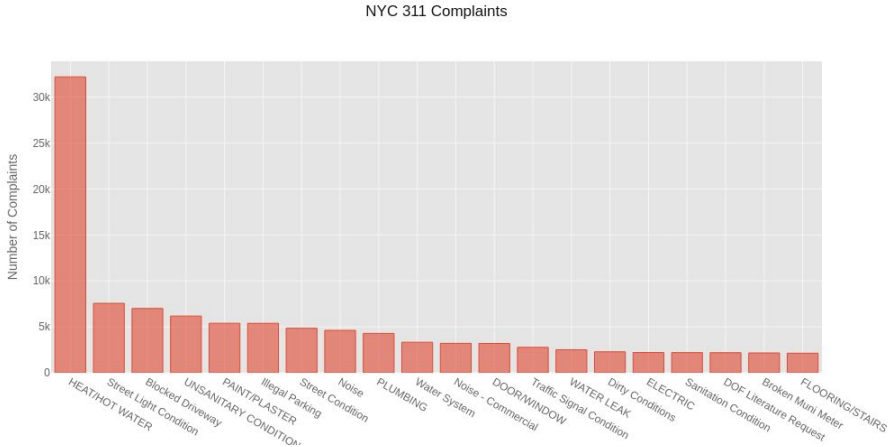
Folium Magic

Fair warning: Not easy to use, but the results look good.

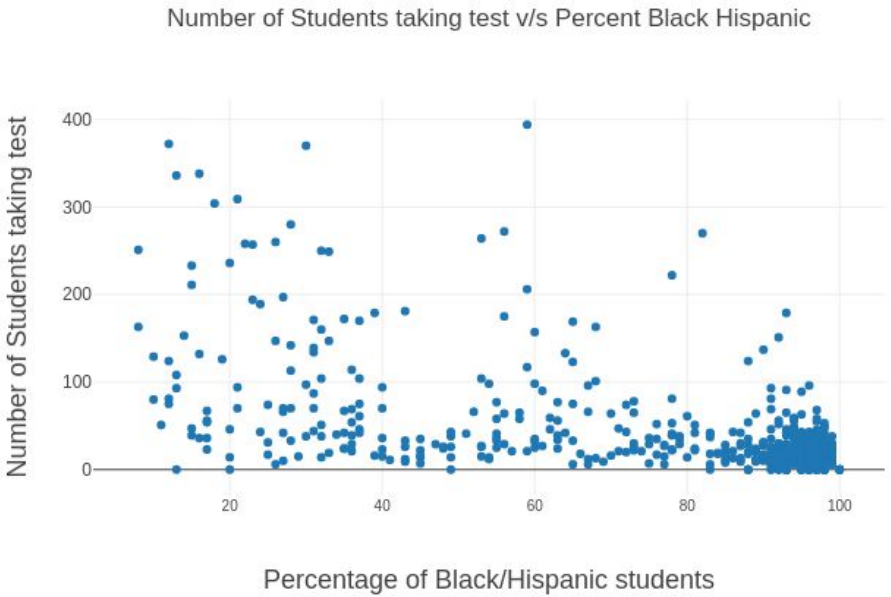
Link

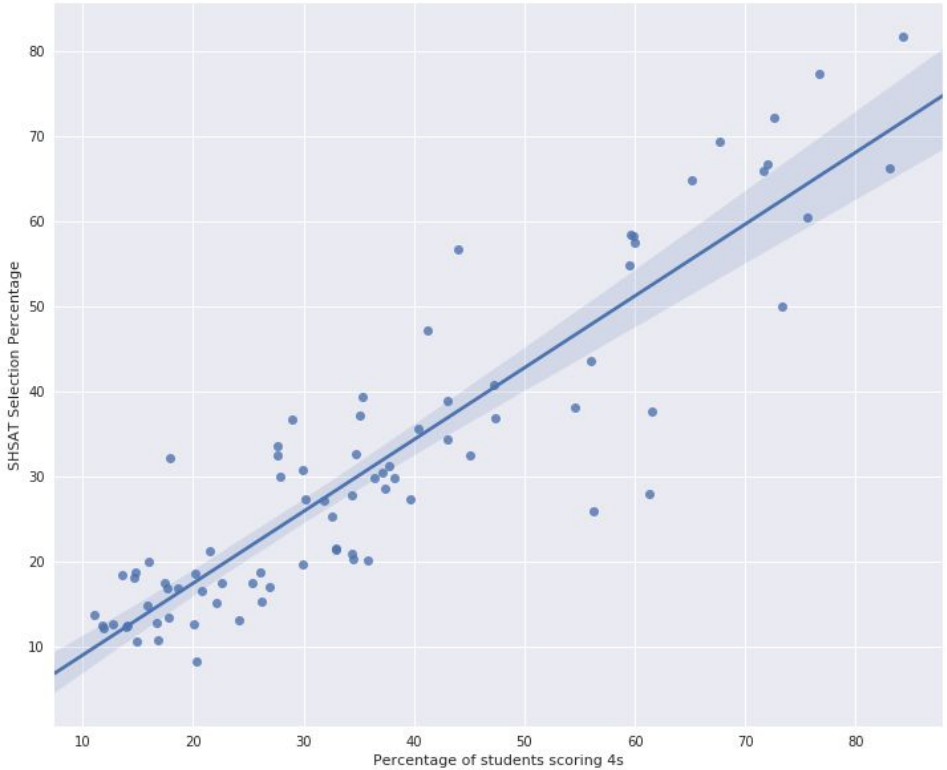






Interactive plots with plotly





Visualizations should be self explanatory

https://colab.research.google.com/drive/1PnQEAirJDHpgtZQV055D63taKf9--YQk

http://localhost:8888/notebooks/Tensorboard%20Visualisation%20of%20Fashion%20Items.ipynb

http://hedwig:6006/#projector

“Mantras” for data scientists

1. Know how to formulate

Formulating the problem is the key.

Focus on the other system constraints that are brought with it.

- Memory / compute size constraints
- Cost constraints
- Class balance
- When will the features be available
- Missing data?
- How reliable the inputs are?
- Kaggle type of problems are not real enough! The real world problems are not well-defined. The onus is on you to define it.

2. Know the context

However important part ML may be for the system, it is not an isolated part

Instead, ML is a part of some system, some product.

Get sense of timing, get sense of what customers want. Accuracy is not everything!

Examples:

Cancer detection

Google search

3. Know the data and domain

Knowing the process of data generation and creation is important. It decides which models we can apply to the problem (more importantly, which ones we cannot!)

Is the training data distribution same as real-life testing distribution?

Having domain knowledge about the field also helps

Examples:

- All projects at Infocusp

4. Know the layout

- No single ML tool offers everything necessary. There are pros and cons.
- Fixating on a single tool/technology can be disastrous.
- Exploring other options stretches how far one can go with ML

5. Know the maths

- Knowing how to "drive" the tools is not enough!!
- Knowing the maths and theory behind the algorithms enables you to make informed decisions.
- The deeper you know maths, more chances there are to reach to an optimal state in a quick time.
- Probability
- Statistics
- Numerical Optimization
- Linear and matrix algebra
- Functional analysis
- Bit of calculus

6. Be a good communicator

- ML and data science is extremely interdisciplinary.
- You will meet people with lack of belief and lack of knowledge.
- Eventually, data mining and machine learning practitioners report to someone.
- Important to back your claims with sound analysis.
- Important to present the findings effectively, being a good visual communicator helps.

7. Spend time knowing the data

———

The model is only as good as the data is.

- You cannot change the raw data, BUT
- you can clean it.
- you can transform it into powerful features.

There is no theoretical limit on how far the feature can take your problem further.

Examples :

- Speech processing
- Face recognition
- Face detection

- Deep learning is changing the landscape … but only partially.

8. Occam’s razor

– – –

- **Occam’s Razor : It is your friend!**

- Clarification : It does not mean, complex algorithms should not be used!
- It means that simpler models are preferred given all the other things being the same.

- Understanding bias variance tradeoff helps.

Other 8 points - tomorrow

Version control systems

- Prevent multiple copies of source code
- Merge changes from different users seamlessly
- Prevent code from breaking repeatedly
- Always have a working copy in case of inadvertent changes
- Changes are modular and you know whom to blame :)
- SVN, git most popular
- Github based on git

GitHub

- An excellent platform to showcase your projects
- Get suggestions/ ideas from the community
- Make sure your work is being used by the people and it comes in handy (Write about it)
- Look at pending issues in popular repositories
- Answer queries, raise queries yourself when stuck
- My advice - use it as much as you can

Contact Details

Falak Shah
falak@infocusp.in
http://falaktheoptimist.github.io