Data Science workflow and tools

ZEMANTA DATA SCIENCE SUMMER SCHOOL 2018

Data Science

- Defining data science
- Importance of data science
- Basic tools

Definition

- An interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms
- A concept to unify statistics, data analysis, machine learning and their related methods in order to understand and analyze actual phenomena within data
- Breaks down to:
  - Data analysis
  - Modeling/statistics
  - Engineering/prototyping

Importance of data science

- We are being overloaded with data
- Some might call this load of data "BIG DATA"
- A human brain can't even fathom the scale of big data
- So we introduce a data scientist:
  - Who is a mixture of: a statistician, an analyst and an engineer
  - Someone who can wrangle the data and through that add value the project or organization he is working for

Tools

- Python
  - Most used language for data science
  - Has optimized libraries for data science
  - Has large active community
  - Easy integration with existing infrastructure
- Git
  - Version control

Workflow

- Acquiring data
- Cleaning
- Analysis and visualization
- Feature engineering
- Models/offline tests
- Model scoring
- Online AB tests
- Assessment and presentation of results

Acquiring Data

- Structured databases
- Logs
- Web portals that offer open data like Kaggle
- Web scrapping

Data Cleaning

- Data science is 80% cleaning data and 20% modeling
- The acquired data is almost always messy
- It's missing values
- Wrong type format like floats are strings with , instead of .
- Wrong formatting on strings like dates
- Extra white space

Analysis and Visualization

- Once the data is clean we want to collect some insights
- Normally you can't just glance at the data and know what it is about
- Aggregations (max, min, mean, median, deviation)
- Plotting and calculating distributions
- Finding correlations between features

Feature Engineering

- We have some insight into the data now
- Feature evaluation
- Picking the right features
- Transforming the features
  - Logarithmic scale
  - Calculating their roots
  - Squaring them
- Feature combinations

Models and offline tests

- At this stage we have the dataset prepared for modeling
- You can choose:
  - Different machine learning models
  - Different sets of features
  - Different parameters for the machine learning models
- Offline tests
  - Train the models on the same set of training data
  - Test and evaluate the models

Model scoring

- Compare the results of the offline test
- Choose the proper metric for the data you are working with
- Pick out the best model and it's time for an AB test

AB

- Have two models:
  - Current production model
  - New model
- Always train the algorithms on the same data
- Strict isolation
- Change only one variable a time
- AAB test, when both As converge you can safely assume that you have gathered enough data to stop the AAB test
- Evaluate how each model did in your production environment

Assessment and presentation

- Assess the performance of the new model:
  - Efficiency
  - Accuracy
  - Efficiency/Accuracy
  - Business impact
- Make a presentation:
  - Show the differences between the efficiency and accuracy of the compared models, using plots is a good choice
  - Give an explanation for the results
  - Explain why you would or wouldn't change the current production model

Machine learning

- Subset of artificial intelligence
- Often uses statistics to give computer the ability to “learn”
- Focuses on construction of algorithms that can learn from and make predictions on data
- Machine learning devises complex models and algorithms which are used to make predictions and classifications

Git

- Git explained
- Git basics

What and why?

- A form of version control
- It allows you:
  - To keep track of when and what you did
  - Undo any changes
  - Easier collaboration with other people
  - Repeatable reserch

Basics

- Get the repository from Git:
  - git clone git@github.com:LukaAndrojna/zemanta_datasci.git
- Create local branch:
  - git branch –b <branch_name>
- Check, add and commit the changes you have made:
  - git status
  - git diff
  - git add
  - git commit
- Push your changes to your remote branch
  - git push

Virtual Environments

- Reasoning behind the use of virtual environments
- Set up
- Usage

Why we should use virtual environment?

- Allows us to:
  - Isolate python environments for our projects
  - Use different versions of the same library in different environments
  - Keep a cleaner distribution of python on our local machine
  - Test modules without any external packages interfering
- Helps with reproducible research
- It creates a server-like environment, where we might have only a clean installation of python
- pip + requirements.txt

Set Up

- mkdir ~/.virtualenvs
- pip install virtualenvwrapper
- vim ~/.bashrc and add:
  - export WORKON_HOME=$HOME/.virtualenvs
  - export PROJECT_HOME=$HOME/Documents/py
  - source /usr/local/bin/virtualenvwrapper.sh

Usage

- Make new environment:
  - mkvirtualenv <env_name>
- List existing environments:
  - workon
- Selecting the environment you want to work in:
  - workon <env_name>
- Exiting the environment:
  - deactivate

Jupyter Notebooks

- What and why
- Set up
- Demo

What they are and why we use them

- A jupyter notebook is like an IDE, consisting of cells within which you can run snippets of code
- Pros:
  - Notebooks are great for experimentation and research
  - Literate programming (development style)
- Cons:
  - Python's scoping is a mess which can get amplified with the use of jupyter notebooks
  - Enables bad coding habits

Set Up

- mkvirtualenv summer
- workon summer
- pip install -r requirements.txt
- ipython kernel install --user --name=summer

Numpy

- Adds support for large multi-dimensional arrays and matrices
- Has a collection of high-level mathematical functions that can be applied on these arrays
- Comparable functionality to MATLAB
- Is based on C, thus providing a more efficient way to work with data
- Is the base for Pandas

Pandas

- What and why
- Basics
- Demo

Pandas

- Python Data Analysis Library
- Data wrangling made easy
- Can natively read csv and tsv files or even SQL databases
- The data is then stored in a DataFrame (df), which resembles a table in Excel
- Once your data is in a DataFrame, you can:
  - Clean it easily
  - Use aggregation functions on it
  - Make selections
  - Make joins with other DataFrames

Basics

- import pandas as pd
- Reading the data:
  - pd.read_[csv, table]()
- Aggregations:
  - df.mean(), df.corr(), df.count(), df.max()
  - df.min(), df.median(), df.std()
- Dataframe descriptions
  - df.describe(), df.isnull()

Matplotlib

- What and why
- basics

What and Why

- Matplotlib is used to draw plots and make visualizations
- A plot is worth a thousand models
- With plotting we get a visual representation of the data we are working with
- We actually see how our models are performing if we are fitting lines in 2D or plains in 3D space
- Use visualizations to present findings

Basics

- Basic scatterplot with a regression line
  - plt.scatter(x, y, color='black')
  - plt.plot(x, regr.predict(x), color='blue', linewidth=3)
  - plt.ylabel('area per person')
  - plt.xlabel('GDP')
  - plt.show()

Help

- Google is always your friend and a good place to start
- Stack Overflow will provide you with examples and solutions for your warnings and errors
- Data science blogs usually provide a good starting example that you can later expand or modify to fit your needs
- Documentation for libraries is a must read, so you know what, where and how to use it, when the opportunity presents itself

Next Steps

- https://www.edx.org/course/python-for-data-science
- https://www.edx.org/course/data-science-essentials
- https://www.coursera.org/learn/machine-learning