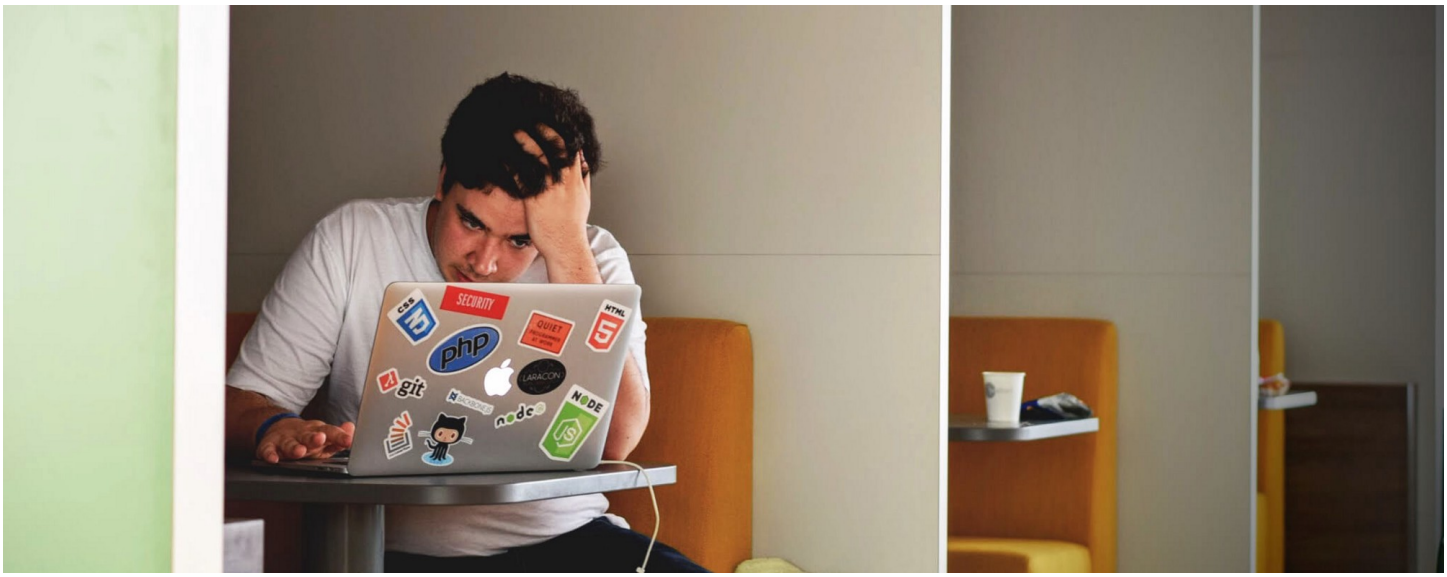
Denis Semenenko Follow
Solves problems at Statsbot
Jun 22 · 5 min read

Data Scientist Resume Projects

Machine learning problems set to build a data scientist CV without work experience



Data scientists are one of the most hirable specialists today, but it's not so easy to enter this profession without a "Projects" field in your resume. You need experience to get the job, and you need the job to get the experience. Seems like a vicious circle, right?

Statsbot's data scientist Denis Semenenko wrote this article to help everyone with making the first simple, but yet illustrative data science projects which can take less than a week of work time.

> *The great advantage of these projects is that each of them is a full-stack data science problem.*

This means that you need to formulate the problem, design the solution, find the data, master the technology, build a machine learning model, evaluate the quality, and maybe wrap it into a simple UI. This is a more diverse approach than, for example, Kaggle competition or Coursera lessons (but they are quite good too!).

Keep reading if you want to improve your CV by using a data science project, find ideas for a university project, or just practice in a particular domain of machine learning.

Spam or Ham

Spam lives wherever it's possible to leave messages. One of the classic data science problems is a spam detection. You can train a model for detecting spam emails, spam messages, and spam user comments to hide them in browser.

A machine learning engine defines spam based on the probability of meeting such words as "sale" and "buy" in spam messages. As a result, you can get a working prototype of AdBlock in about one week.

ML problem: text classification
Algorithms: naive bayes, linear classifiers, tree classifiers, whatever-you-want classifiers
Technologies: sklearn, nltk, scrapy
Data: sms spam dataset, e-mail spam dataset , youtube comments spam dataset
Implementation: browser extension
References: AdBlock, Adguard
Guides: How To Build a Simple Spam-Detecting Machine Learning Classifier, Getting Started: Building a Chrome Extension

Not Hotdog

Not Hotdog is a viral app from the Silicon Valley series which can recognize hotdogs and non-hotdogs in photos.

You can train an image classifier for whatever you want by using social networks or google images to collect a dataset (using hashtags or search queries).

Training of a neural network from scratch requires a lot of train samples and computing time, so it is better to use a pre-trained network (this approach is called transfer learning).

ML problem: image recognition, image classification, transfer learning
Algorithms: convolutional neural networks
Technologies: keras, lasagne, Instagram API(or external libraries e.g. Instabot)
Data: use Instagram API and hashtags to collect dataset
Implementation: mobile app
References: Not hotdog
Guides: Transfer learning using Keras, Building powerful image classification models using very little data

Netflix movie recommendations

Recommender systems are necessary for large companies like Google or Facebook, because it is valuable from a perspective of revenue and engagement (Facebook ads, Youtube recommendations).

Data scientist beginners are able to get practice in this sub-domain of data science and build their personal movie recommender system.

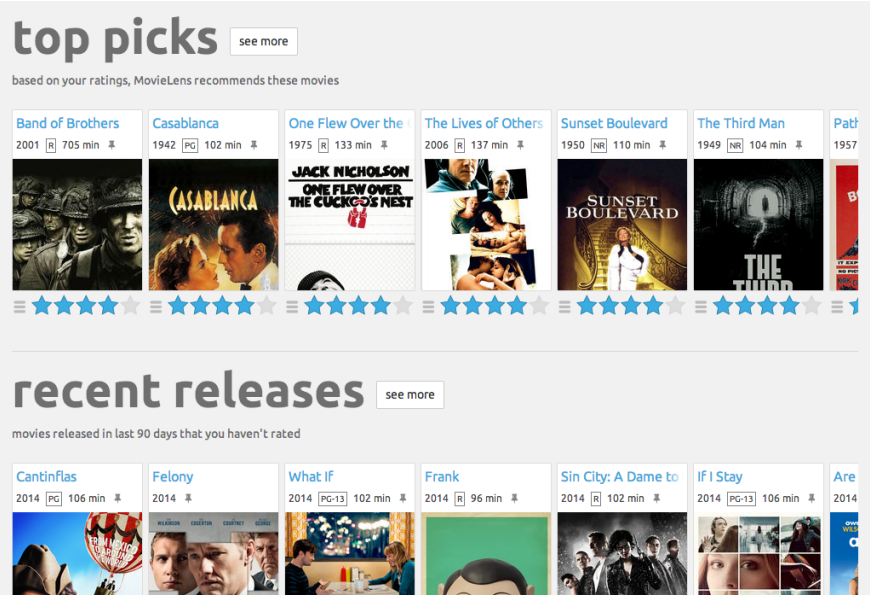


Illustration: movielens.org

ML problem: recommender systems
Algorithms: dimensionality reduction, collaborative filtering, classification algorithms
Technologies: sklearn, vowpal wabbit
Data: Netflix prize dataset, MovieLens dataset
Implementation: web application
References: Jinni, MovieLens
Guides: Quick Guide to Build a Recommendation Engine in Python

Original Snapchat lenses

If you are good at graphical design you should try to create your own custom Snapchat lens. Any social network has such things in their arsenal—Instagram, Facebook, and Snapchat.

Gif: https://support.snapchat.com/en-GB/a/lenses1

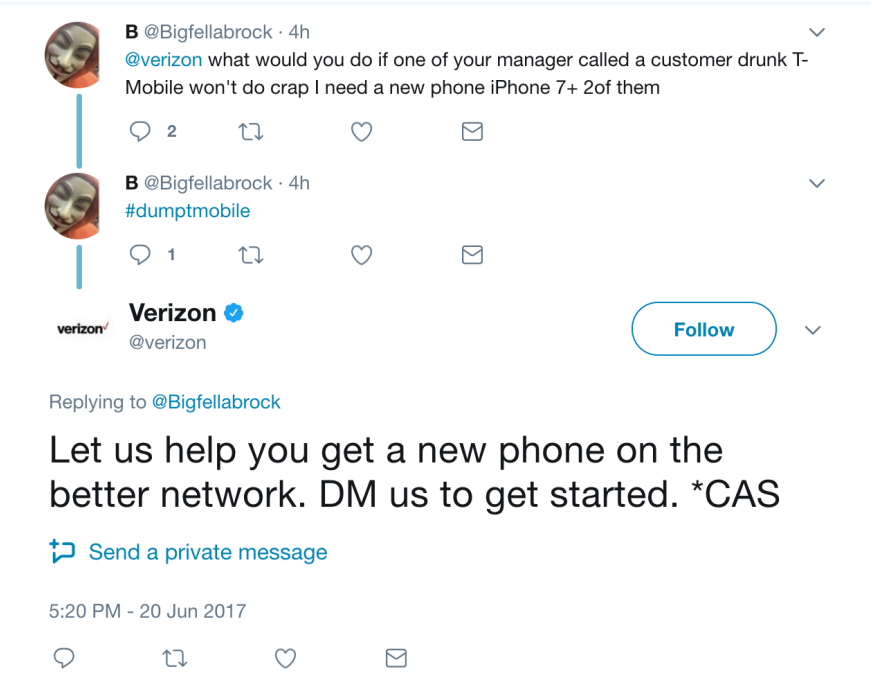
Lenses detect key facial points to show where the borders of the lips, eyes, nose, and face are. Then, ML engine builds a mask matching the face correctly even if you're moving (for example, using openCV or any graphic library).

ML problem: image recognition, face detection
Algorithms: convolutional neural networks, facial keypoints detection
Technologies: dlib, openface, keras, openCV
Data: Facial keypoints detection dataset
Implementation: mobile app
References: Snapchat, Instagram

Guides: Facial landmarks with dlib, OpenCV, and Python, Build a Simple Camera App

Twitter stream

A lot of companies monitor mentions from their customers on the internet to react to the negative ones ASAP. For example, T-Mobile and Verizon need to respond to negative tweets fast and find out what the problem is and how they can solve it.



Everyone can make this project using a convenient Twitter API and sentiment analysis algorithms to detect such tweets in the whole stream.

ML problem: sentiment analysis
Algorithms: sentiment analysis
Data: Twitter API
Technologies: nltk, spaCy
Implementation: web app
References: Twilert, Tweetreach
Guides: Twitter sentiment analysis using Python and NLTK

Tennis betmaker

Professional betmakers always look for profitable bets. Tennis is a good kind of sport for predictions due to large amounts of statistical

data, and sometimes bookmakers make mistakes.

Data scientists can put together websites with match history and players' information, build a prediction model, and compare the results with bookmaker's evaluations.

The goals is to find a gap between ML prediction and a bookmaker's evaluation, which gives a chance to win. It's a good data scientist problem for feature engineering!

ML problem: classification
Algorithms: classification
Data: atpworldtour.com
Technologies: sklearn, scrapy
Implementation: web app
References: olbg.com, verifiedbets.com
Guides: Machine Learning for the Prediction of Professional Tennis Matches

Stock price prediction

A popular application of ML is time series prediction. A data science engine can predict exchange rates and stocks, so traders or bots can gamble based on these predictions.

If you choose this problem, you'll find out that it's easy to get such data and practice on it. This sub-domain is derived from econometrics and classic machine learning, so you should be ready to explore a statistics method.

ML problem: time series prediction
Algorithms: ARIMA, regression
Technologies: sklearn, prophet, scrapy
Data: Quandl
Implementation: web app
References: financeboards.com
Guides: An Introduction to Stock Market Data Analysis with Python

. . .

I hope you enjoy these toy problems in data science! Using them, it's simple to improve your data scientist resume and get a high-demand job.

If you know some more relevant ML projects, please share them in the comments below and I will include them in an article. And of course, recommend this post if you find it helpful for you or other data science lovers.

YOU'D ALSO LIKE: