Open in app

This account is under investigation or was found in violation of the Medium Rules.

uday sai

4 Followers About

The art of being a Full-Stack Data Scientist

uday sai Apr 1, 2020 · 7 min read

Image Source: Photo by Benjamin Davies on Unsplash

Data Scientist is touted as one of the hottest job areas in the 20th century and it's no wonder that many of the budding engineers are aspiring to be a Data Scientist. In spite of the zeal and enthusiasm many are finding it difficult to get a Data Scientist job in the industry. This article is intended towards beginners who are aspiring to be a Data

Let’s deep dive into the typical pipeline that is often implemented in a real-world data science projects. The delineation of the steps are as follows:

> *Understanding the domain and setting up benchmark metrics*
>
> *Data collection and EDA*
>
> *Data Cleansing and Feature Engineering*
>
> *Model Development and Performance Evaluation*
>
> *Model Deployment*
>
> *Post Deployment analysis*

Let us explore each step on at a time:

Understanding the domain and setting up benchmark metrics:

To start with a data science project in a real-world it is very important to understand the problem statement and the domain in which we are conducting our analysis. For example let us consider a popular use of ML in finance industry for credit card fraudulent detection (I will be using this example through out this article), in this case it is very important for us to understand the variable and get a better idea how our solution is compatible with the business model. This requires assessing the solution in terms of how effective the solution is when compared to existing system in terms of manual effort it reduces, revenue it brings, cost it incurs. Following this step we need to set up an objective let us say to detect the credit card fraud cases with an efficiency of 80% and mark it as an benchmark and work towards achieving it. Besides we also need to consider other functional requirements of latency of the prediction use case, number of transactions per second etc… This marks up in strategizing a plan and executing it.

Data collection and EDA

On contrary to the projects in academia where most of the data comes from flat files, in real-world the data always comes from disparate sources like relational databases, assists in interacting with databases. However, this data collection requires SQL skills like writing optimized queries and not to drain up the computing resources. In addition to databases and data lakes sometimes extracting the data over web using Beautifulsoap, getting the data by interacting with API's is also very handy. Let us consider our illustration of Credit card fraud detection there might be a case where all the customer related data is stored in PostgreSQL and all the transaction data exists in a hadoop ecosystem. In this case we need to fetch the data from multiple resources.

Before we move on to a crucial phase in fact most important phase of Data Science project I would like to quote that in many of the projects that beginners work, this phase is often undermined. I would like to emphasize that this phase requires more attention than any other phase in project cycle. Let us hop in actual aspects of phase, data cleansing is the part in which we will structurize the data by applying business logic and minimizing the noise. Tracing back to our credit card fraud detection phase there is a possibility to have naming disparities of a customer name in multiple resources, we need to reconcile these issues before we proceed to next step. Apart from this applying some of the dataframe operations like melt, concatenate, merge, groupby, pattern matching using regular expressions, using list comprehensions and lambda functions and defining custom functions are some of the worthy skills to have in our repertoire.

Data Cleansing and Feature Engineering

Following the data collection we need to perform some exploratory data analysis by both uni-variate and multi-variate analysis like generating histograms to check the spread of the variable, scatter plots to find the interaction between variables and applying df.dtypes to check data types, df.describe() to check some common statistics and df.value_counts() to check the number of unique values especially of a categorical variable. Besides we need to check for Missing values, distinct values of categorical variable, detecting anomalies etc…

The step of feature engineering is like an experimentation phase where we try to combine features and get the maximum explainability out of these features. The objective is to have maximum explainability with minimal number of features, being said that don't mistake it with dimensionality reduction techniques like SVD, PCA. These

Simple imputer to KNN imputer and different encoding schemes especially for nominal categorical variables like hashing technique, deciding on categories based on frequency of the unique values of variables.

These techniques are of at most importance from a real-world project stand point as these impact the performance of the model and be a cog in wheel in achieving the business objective that we bench-marked earlier. In our credit card fraud detection case we can consider zipcode as a nominal categorical feature which has lot of unique values and using hashing encoding might save us as traditional one-hot encoding will blow up columns in our dataset. There may be a situation where we need to come up with custom logic to deal with these scenarios. I not only urge people to practice many feature engineering techniques and data cleansing methods but also to be creative in generating them.

Model Development and Performance Evaluation

The next step is to build a model, with the level of sophistication available in sklearn the model building is one click away but what stands out is coming up with the hyper-parameters using methods like Grid Search CV, Random Search and building robust test sets using different variants of cross-validation technique. We can experiment with these hyper-parameters and make sure to reconcile the bias vs variance trade off by examining the validation and learning curves. This is an important aspect of the analysis.

The performance evaluation is not always confined with accuracy and we need to examine the other metrics like confusion matrix, precision, recall, RoC Curve ,Area Under Curve (AUC). In our credit card fraud detection statement confusion matrix is of paramount importance as most of the datasets will be highly imbalanced and we need to examine confusion matrix to build a better model. Besides we need to come up with the model which has a better recall than precision as we need to capture maximum number of fraud cases. Based on the performance metrics we may iterate over the earlier steps and modify our approach to build a better model.

Model Deployment

Many people ponder over a series of questions like why do data scientist need cloud and CI/CD skill set. This step is the answer for such questions. On achieving the benchmark python) the model and build a web application using flask or django based on the requirements. The convention is to use flask for small and relatively medium datasets and django for large models.

In order to overcome the dependency challenges like installing libraries we containerize the models using docker and use container orchestration frameworks like kubernetes to address like reproducibility, scalability and zero downtime deployment. In our credit card fraud detection case we may need to deploy it on a real-time data and authenticate the transaction with in a millisecond of time. This can be achieved by addressing the functional requirements that we defined in step 1 like addressing the 500–600 transactions per second and our model should be deployed in zero downtime environment by leveraging kubernetes abilities. Additionally tools like gunicorn and ngnix can be used to scale, parallalize the workload and for load balancing respectively. Eventually the model can be deployed in any of the cloud environments like AWS, Azure or GCP.

Post Deployment analysis

After deploying the model in the cloud environment we need to inspect the model for a good of time in order to see if the model tends to deviate from the structure of the data. This scenario appears in many of the real-world projects and needs to be examined with close attention. This is technically called **“Concept of Drift”**

Path Forward

I recommend audience to follow these steps when they are trying to accomplish data science projects. Please consider collecting data from multiple resources like news articles, twitter or any other trusted repository, this gives an immense opportunity to work on some advanced cleansing and feature engineering techniques.

References:

Real World Machine Learning Pipeline (ML Engineering)

Few months back I had put an presentation highlighting disconnect between academics/popular courses and enterprise in…

medium.com

Photo by Benjamin Davies on Unsplash

Data Science Full Stack Python3 Machine Learning