Tech Mahindra

Pitfalls
in Applying

DATA SCIENCE & ANALYTICS TECHNIQUES TO SOFTWARE TESTING

Learn how to safeguard against these Pitfalls

Authored by
Dinesh Velhal & Mohit Saxena

Connected World.
Connected Experiences.

Abstract

In recent years, Data Science techniques is seen to be revolutionizing the Industry sectors ranging from Healthcare, Banking, and Communications to Narcotics. AI based techniques, over the years, have assured substantial outcome based promises where hitherto traditional analytical solutions fell short of delivering desired results.

Experts suggests that by 2020, 85% of CIOs will be piloting AI programs through a combination of buy, build and outsource efforts. There is need and drive to apply these techniques to software testing field. However, software testing poses its own unique challenges, which necessitates careful analysis before you can successfully apply them. This paper outlines some important pitfalls in applying AI/ML techniques and provides insights in avoiding those.

Introduction

Industry studies shows that testing easily takes up about 25-30% of the total project cost on an average. With most organizations adopting the DevOps and CI/CD/CT methodologies, there is even more pressure on testing to be more efficient and align to the pace of these methodologies. This is one key driver for considering AI/ML based techniques to improve testing efficiency.

Steps involved in implementing Machine Learning

Before considering the Data Science techniques, it's important to understand the steps involved in their implementation. Following diagram is based on the CRISP-DM Data Mining standard.

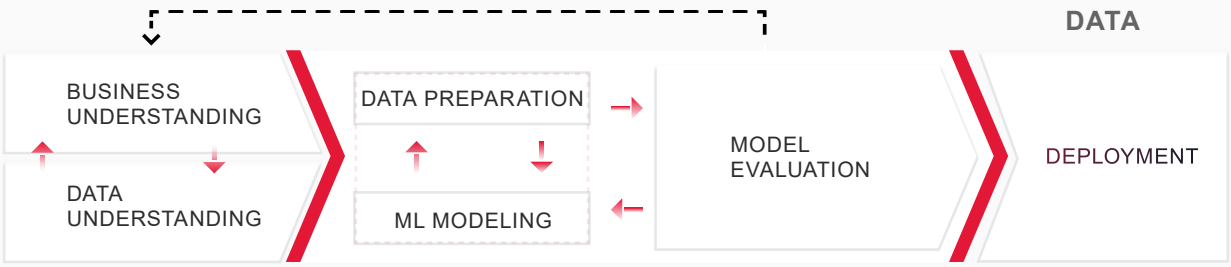

Figure 1: CRISP-DM Process

BUSINESS

- Define Problem / Opportunity
- Gather details *(Assumptions, constraints, risks, resources etc)*
- Formulate the goal / desired outcome

DATA UNDERSTANDING

- Data Acquisition
- Preliminary/Exploratory Data Analysis

DATA PREPARATION

- Address quality issues – *(Clean the data, identify features to use, standardize/ normalize the data)*
- Prepare data for modeling

MODELING

- Determine type of problem and accordingly the suitable ML algorithm
- Build the model

MODEL EVALUATION

- Assess the model performance
- Evaluate the model outcomes against the success criteria
- Select the model or Go back to previous steps

DEPLOYMENT

- Present the model and outcomes
- Deploy the model in live environment
- Monitor the performance

For the sake of simplicity, following section talks about some common challenges/pitfalls by clubbing the steps.

1 Business Understanding | Data Understanding

Challenges

1. Are you solving the right problem?

Many testing problems are due to the incorrect testing tools, processes or techniques. A detailed test assessment can highlight these adversities, and suggest corrective ways to resolve them. It's strongly recommended to first get the basics in place and then apply AI (Artificial Intelligence) techniques to further gain efficiencies if needed. Implementing AI based solutions is costly and should only be implemented to solve the right set of problems (those which cannot be solved using traditional techniques)

2. Lack of sufficient data with sufficient quality?

Analytics and AI solutions are data-intensive. Normally, the more the data, better is the performance. One needs to carefully analyze if one has sufficient data with required quality. Key questions to ask:

- Do we have required data available in our enterprise applications?
- Is the data accessible for use? E.g. the testing release data is owned by the ALM team from a vendor. Are there processes in place to enable access to data owned by various teams?
- Can the data access be automated through data pipelines or accessing it is a manual process? This is key to the successful deployment of the AI solution in production
- Is the data consistent across different teams? This is key to getting consistent results from AI based solutions. E.g. different teams follow different guidelines when it comes to writing test cases or defect reports. If this is the case, then it may reduce the overall effectiveness of AI based solutions

A lot of testing data is human generated E.g. test cases, defects, test specifications. The usefulness of such data can vary from project to project and may require extensive massaging before it can be rendered useful.

In contrast, machine generated data (examples - application logs, system metrics) is much more consistent and can provide consistent results. Data preprocessing can be automated to a larger extent compared to human-generated data.

2 Data Preparation Modeling Model Evaluation

One critical challenge about the modeling is to develop a model that works for variety of data points and keeps working consistently. In general, it is observed that the models that work well under lab conditions fail dismally when deployed in production. Main challenges behind this cause are as follows:

Challenges

1. Data characteristics undergo changes over time?

This is popularly known as the problem of Data Drift. In testing projects, this can happen for variety of reasons – some of them listed below

- Change in the test organization / leadership / team composition – These kind of organizational changes affect the testing process followed in subtle / explicit ways in turn affecting how testing data is generated. This directly impacts the model performance (mostly in adverse ways) which was trained on a snapshot of data taken in the past
- Change in the testing tools
- Major changes in the applications under test – These changes can be technical, architectural or compliance related. In general, the effect is similar to what is described in the previous point.

2. Models are trained on data that don't represent all practical situations in live systems?

This may be caused due to lack of holistic problem understanding or due to lack of access to datasets from various applications, teams or departments. The end result is – the model performs well for some data points but not so well for the other. So it's important that a comprehensive data selection and preparation activities are performed to make the model more general in nature.

An effective way to reduce the impact of these issues is to keep performing the data preparation / modeling activities at regular intervals to accommodate the changes to the data characteristics. Effective monitoring in live environment can also indicate the issues early after the deployment.

When we recently implemented Predictive Analytics solution for a Telecom Testing project, we noticed that the solutions performance (prediction accuracy) would vary based on certain practical release parameters. So we set up a weekly monitoring & review of predicted outcomes and improve the performance using various techniques. Retraining the algorithms with latest release data provided significant boost to the performance.

3 Deployment

Most machine learning articles, papers or tutorials focus on only the data analysis and modeling aspects of the overall process. While these are really important for the success of AI based solutions, they form only a part of the overall story. The less discussed aspect of the AI based solutions is the challenges and overheads encountered when the solution is deployed in production.

While the data preparation and modeling activity is done on a snapshot of data in non-production environment, you benefit from them only when the models are deployed in live. This involves additional engineering steps that may vary in terms of complexity based on what kind of problem you are solving.

This can be as easy as presenting the results to senior management or can get as complex as building automated data pipelines to process and integrate data from various systems and making it ready for analytics in real time. These additional engineering efforts also need to be thoroughly tested before they can get deployed on production.

Thus it's important that, additional efforts need to be factored in when planning for the AI based solution implementation to avoid any delays at the end of the implementation process.

Another deployment related challenge is about the CI/CD and DevOps practices which are mature and efficient for traditional development. For machine learning development, they are still evolving and may take some time before they mature. E.g. For traditional software development, code version control is easily achieved using a range of version control systems. For Machine Learning applications, code and data also need to be versioned. Tools supporting these are still evolving.

Conclusion

AI based solutions can bring great benefits to any testing projects provided following aspects are thoroughly analyzed before the implementation.

- Are you solving the right problem?
- Do you have quality data in enough quantity? Does organization have measures in place to make data easily available?
- Do you have right skills available and planned the right budget for the implementation?
- Above all – do you have a data-centric culture that empowers the AI/Data Science teams to acquire and process the data for solving the business problems?

Mohit Saxena
VP & Global Competency Head
Digital Assurance Services, Tech Mahindra
LinkedIn

Mohit heads the Digital Foundations, ADMS and Testing competency in Tech Mahindra. He has closely worked with clients on Digital transformation and agile implementation which included moving from legacy Waterfall to Agile way of working and creation of self-forming and self-governing teams [Tools, people and process]. Implementation of DevOps [CI/CD/CT] with dynamic environment management is his expertise. His recent passion includes application of Cognitive DevOps, creation of software defined networks using Ansible and Infrastructure as code (IaC) using Chef, Puppet , Dockers along with Kubernetes. He believes NoOps can be a reality

Dinesh Velhal
Practice Head
Digital Assurance Services, Tech Mahindra
LinkedIn

Dinesh has helped various clients in implementing test automation across various geographies and verticals. His software testing experience spans 10+ years and includes Test Management, Automation Frameworks, Agile Testing, Automation Consulting, Test Reporting & Predictive Analytics. He is a Machine Learning Enthusiast and most recently, he has implemented ML-based Predictive Analytics for large testing projects. He has previously worked for 8+ years in software development using Java, .NET, Unix, SQL & Oracle PL/SQL for Telecom and Banking domains.

Tech Mahindra

Tech Mahindra, herein referred to as TechM provide a wide array of presentations and reports, with the contributions of various professionals. These presentations and reports are for informational purposes and private circulation only and do not constitute an offer to buy or sell any securities mentioned therein. They do not purport to be a complete description of the markets conditions or developments referred to in the material. While utmost care has been taken in preparing the above, we claim no responsibility for their accuracy. We shall not be liable for any direct or indirect losses arising from the use thereof and the viewers are requested to use the information contained herein at their own risk. These presentations and reports should not be reproduced, re-circulated, published in any media, website or otherwise, in any form or manner, in part or as a whole, without the express consent in writing of TechM or its subsidiaries. Any unauthorized use, disclosure or public dissemination of information contained herein is prohibited. Unless specifically noted, TechM is not responsible for the content of these presentations and/or the opinions of the presenters. Individual situations and local practices and standards may vary, so viewers and others utilizing information contained within a presentation are free to adopt differing standards and approaches as they see fit. You may not repackage or sell the presentation. Products and names mentioned in materials or presentations are the property of their respective owners and the mention of them does not constitute an endorsement by TechM. Information contained in a presentation hosted or promoted by TechM is provided “as is” without warranty of any kind, either expressed or implied, including any warranty of merchantability or fitness for a particular purpose. TechM assumes no liability or responsibility for the contents of a presentation or the opinions expressed by the presenters. All expressions of opinion are subject to change without notice.

www.techmahindra.com

connect@techmahindra.com

www.youtube.com/user/techmahindra09

www.facebook.com/TechMahindra

www.twitter.com/Tech_Mahindra

www.linkedin.com/company/tech-mahindra