Technology Stack for AI Prototyping
# INTRODUCTION

2. THE IMPORTANCE OF AI PROTOTYPING

3. EVOLUTION OF TECHNOLOGIES FOR AI PROTOTYPING / ¿WHERE DO WE COME FROM?
   - 3.1. The Almighty Python
   - 3.2. The Unbeatable Cloud Providers
   - 3.3. Streamlit: The fastest way to build data apps

4. THE AI PROTOTYPING TECH STACK

5. A GOOD EXAMPLE HAS TWICE THE VALUE OF GOOD ADVICE

6. BUT HOW WE SCALE ALL THIS?

CONCLUSIONS

REFERENCES
INTRODUCTION

In the last decade, the popularity of Artificial Intelligence (AI) has massively increased and has infiltrated into our every-day lives up to the point that we are not surprised anymore by the recommendation of related products before paying in an e-Commerce site, nor by the daily interaction with voice assistants. In this new technological era, all companies must take advantage of AI and understand how to embed it into new business models, services, and interactions with their customers, as these expect brands to deliver relevant value through differential experiences.

At NTT Data, we detected that since AI services require a lot of experimentations to ensure the expected results and the fact that this type of technology is not yet very mature, many clients prefer to have definite Proof of Concepts (PoCs) to validate AI use cases before investing more resources into a larger scope project. At the same time, in the last few years, several disruptive technologies have appeared that, if wisely combined, allow Data Scientists to rapidly build end-to-end AI prototypes. At NTT Data, we have been testing and investigating these technologies and have built a Python-centric AI Prototyping Technology specifically for that purpose.

In this guide, we present the need of prototyping AI projects, expose the game changer technologies that allow the rapid and agile prototyping of this kind of projects and introduce our very own AI Prototyping Tech Stack. Finally, we exemplify the use of this technology stack with a valuable use-case for the Insurance sector.
Tech players are setting new rules on the definition and delivery of new market value proposals, embedding AI to deliver outstanding customer experiences. We are currently seeing the rise of the new digital era and in the coming years, all digital experiences and services will as well be intelligent.

Let’s say we are an insurance company that has recently become aware that we need to embed AI into our business models, services, and interactions with our customers if we want to thrive in this new technological era. But… where do we start? We know we have tons of customer data, high skilled IT professionals and brand-new IT infrastructure and so on but we do not have a clear AI-related business-valuable use-case, let alone its guaranteed success.

Regardless of the ideation process, should we start devoting whole interdisciplinary teams to produce a final AI product/service version?

Recent reports [1] show that around 87% of Machine Learning models never make it into production. Lack of data, difficulty in interdepartmental communication and lack of cross-language are some of the main contributors to the final failure in the lifecycle of these models. In addition to these reasons, the fact that AI use-cases are not always correctly validated by different levels of professionals at early stages and that the model’s outputs are not always as expected result in the majority of cases in a large waste of resources.

And that is when AI Prototyping enters the scene. With a tangible prototype, we can rapidly validate that a use-case is viable from a technological point of view and show its business value to sponsors and any other relevant stakeholders even at early stages of experimentation, thanks to an effective visual language. Those to aspects are key to guarantee the success of an AI product or service.

But what is Prototyping anyway?

Prototyping is an experimental process where teams implement ideas into tangible products, capturing design concepts and testing on users. With prototypes, designs and products can be refined and validated so that only the right products are released. In the context of AI, we understand prototyping as the end-to-end conception and development of services and products embedding AI.

Benefits of AI Prototyping

Many are the benefits of AI Prototyping, but we go deep on the following six ones:

**BUSINESS VALUE**

Data Scientists will have the tools and the know-how to demonstrate to business sponsors the value that lies in the data and can be leveraged through AI. Prototyping will create more efficient feedback loops, as the visual representation of the power of the AI models will inspire subject matter experts in order to formulate recommendations.

**CULTURE OF EXPERIMENTATION**

The successful implementation of a prototyping methodology allows AI to permeate all business areas and processes, showing their potential (or not) business value rapidly and reducing the frictions that lengthy, unsuccessful projects may generate. As the value of AI can be more easily grasped, more stakeholders will understand the importance of gathering quality data to fuel the prototypes that may transform the organization.

**COST**

With prototyping, the use cases are validated before being deployed. This investment prevents costly errors in advance because project success is guaranteed.
Evolution of Technologies for AI Prototyping / Where do we come from?

Some years ago (but not that many years ago), AI Prototyping was far from being a fast and agile process. It took months and different professional profiles to create a tangible prototype to validate an AI use-case. Once that prototype was developed, and in the few cases where the prototype ended up being successful, the prototype resulted in the final product for the number of resources dedicated, so it is not sure if the term prototyping applied in the first place.

In most cases, the backend part of the prototype simply consisted of AI models built from scratch, which require a high amount of expertise, data, and time. For the front-end side, experienced front-end developers (experts on web development and UX/UI) needed to create a portal where the prototype could be showcased. Those developers should constantly communicate with the Data Scientists developing the backend to integrate both ends. Additionally, an IT architect should develop an infrastructure where the portal should be hosted.

Over the last few years, some new technologies have emerged that have ended up playing as game-changers in the world of AI prototyping. These technologies end up covering the whole full stack of the development process: the backend now can consist of combination of different ad-hoc fine-tuned models, AutoML or 3rd party APIs, while the front-end part is covered by technologies such as Streamlit. With the adoption of these technologies, a single Data Scientist can quickly create, securely deploy, and share a complex AI use-case, which results into more prototypes being implemented, higher speed-to-market and quality and so on.

We believe that there are three pillars that sustain this change of scenario, and we will following discuss them in detail:

3.1 The Almighty Python
The main thing that all these technologies have in common is that they are based on the open-source Python programming language, which has become the most popular programming language for Data Science mainly due to the following reasons:
Nowadays, cloud computing implies the delivery of different services through the Internet, including data storage, servers, databases, networking, and software. The term “cloud computing” was coined more than 20 years ago [2] and since its beginning it became widely used as it introduced the concept of Software as a Service, with companies such as Salesforce having a great impact on the overall industry. However, it was until 2006, with the launch of Amazon Web Services (AWS) where modern cloud computing was created and popularized. Since then, other important providers such as Google Cloud or Microsoft Azure have emerged and the technologies of cloud computing have played a very important role on the accessibility of AI, acting in most cases as an effective catalyst.

The main contributions of cloud computing in AI can be split into the following three subsections, that resemble the different cloud delivery methods:

- **ML Infrastructure**: Provision of a ready-to-use and pay-as-you-go powerful and specific infrastructure such as CPU, GPU, disk memory, networks and so on.
- **ML Services Platform**: Delivery of ML platforms (such as AWS SageMaker) that provide a scalable platform to build, train and deploy ML models quickly, including a large variety of built-in algorithms and functionalities.
- **High-level AI Software Solutions**: Pre-trained AI models that can be easily consumed by an HTTPS API. In the recent years, with the apparition of AutoML, these endpoints can even be customized, so with little effort and practically no experience in AI, custom AI models can be easily built and consumed.

Given the conjunction and rapid evolution of these three cloud-powered AI delivery methods and their wide coverage of the AI spectrum (Computer Vision, Natural Language Processing, Forecasting, Reinforcement Learning and so on), the complexity of the use-cases that can be derived from its use is rapidly increasing. While some years ago it would take a data scientist (DS) months to create a simple use-case that included a ML classification model, nowadays the same DS can implement a complex AI use-case by combining language and vision 3rd Party APIs with fastly trained custom ML models and so on. Later in this guide we will give an example on how to prototype these kinds of complex use-cases.

Seeing the speed at which these cloud giants evolve, one may ask herself how to keep up. Or perhaps even how to get started. For those two tasks, we recommend taking a look at the following list:

- Official and unofficial (Udemy, Udacity and so on) training and certifications help learners build credibility and confidence by validating their cloud expertise with an industry-recognized credential.
Technology Stack for AI Prototyping

3.3 Streamlit: The fastest way to build data apps
The open-source Python library Streamlit was created by Adrien Treuille, a former employee of Carnegie Mellon University, Google, and Zoox. It was conceived with the goal of facilitating the task of easily showing the work done by ML engineers and data scientists, without having to fall into developing web apps that would end up being unmaintainable. Since its launch in 2018, the start-up developing Streamlit has raised more than $62M in different financial rounds [3] and has reached almost 20k GitHub stars (and counting) [4], which means that Streamlit has come to stay.

Moreover, Streamlit allows their users to easily create Custom Components, additional modules that extend the tools basic capabilities. Starting from an officially provided template, developers can write JavaScript and HTML code that is easily rendered in Streamlit apps to fulfill their specific use-case needs. Some examples of this kinds of components that result really useful are a component to embed a code editor in your app, a simple component to display annotated text or a component for easily cropping custom images.

Given all of this, it is no wonder that over 50% of Fortune 50 companies such as Apple, Uber or IBM (as they claim in their website) are currently trusting Streamlit as their main tool for rapidly building and sharing data apps.

For the last few years, at NTT DATA we have been working on multiple client projects and Proof of Concepts (PoCs) related to AI and have experimented with different technologies with the goal of reducing the time for building AI prototypes and increasing the speed of experimentation.

Based on those experimentations, we have built (and are continuously building) our own AI Prototyping Technology Stack, in which we leverage the capabilities that Python offers, using it as the main programming language, along with a set of its extensions, allowing the Data Scientist to have a wider coverage of the prototyping studio, reducing the complexity on the type of teams and delivery methods.

On the bottom of the Tech Stack, there is a Backbone layer that combines the different technologies that provide a fast, secure, and containerized infrastructure to deploy and be able to easily share the prototype applications. This includes the different cloud vendors for providing compute instances, NGINX for the authentication and securing of the application and Docker for containerizing it so it can run on top of every OS.

In brief, by just adding some lines to your already existing Python code, you can instantly obtain a high-performing data app in your browser that is easy to deploy and is responsive for all types of devices.

Streamlit embraces Python scripting and treats widgets as variables, which means that creating an interactive layout becomes a really easy task. It also reuses data and computation as well as adding the ability to deploy instantly.

• Virtually (or physically, if lucky) attend the massive conference that each of the main cloud vendors organize towards the end of each year, where they present keynote announcements, training opportunities and thousands of technical sessions. Currently, the main conferences are AWS re:Invent, Google Cloud Next and Microsoft Ignite.

• Other resources such as newsletters (where the next events, webinars and launches are announced), specific blogs and forums, and even streaming Twitch channels are currently available to broaden and deepen your knowledge of these technologies.

The AI Prototyping Tech Stack

04
On top of the backbone layer, there is the **Data Acquisition layer**, where we rely on beautiful-soup and Selenium for obtaining information from static and dynamic websites respectively. We also include the Requests library to easily send general-purpose HTTP requests.

Next, we find the **Data Analysis and Processing Layer**, where we include Plotly for data visualization, which integrates really beautifully with Streamlit and allows to have more interactive and responding visualizations than other similar libraries. On this layer we also find pandas, the essential library for dealing with DataFrames (tabular data) and NLTK and OpenCV for working with text and visual content in that order.

One of the key layers of the stack is the **AI/ML Models Layer**, which includes on the one hand ML frameworks for building custom ML models from scratch or customizing existing ones (TensorFlow, PyTorch and HuggingFace) and on the other the main cloud vendors that provide high-level AI solutions in the form of APIs.

Finally, the other key layer in our Tech Stack is the Web Application Layer, which leverages the capabilities that the previously introduced Streamlit has to offer for rapidly building data apps and FastAPI, a web development framework to easily expose as independent endpoints the custom ML models.

Let’s go back to imagining that we are an insurance company that wants to define a potential prototype for this sector. We have identified a problematic that consists of manually processing the thousands of emails that we receive from customers regarding car damage claims. These contain explanations on what happened to their vehicle together with personal information (e.g., personal addresses, driver ID, email) and some images of the damages. We would like to use AI to identify all the personal information in the text and classify the type of damage from the image.

We consider that before spending lots of efforts and resources into developing a large solution, it would be worth prototyping this use-case in order to validate its technical feasibility and its business impact by sharing it with the C-level executives.

One of the key layers of the stack is the **Web Application Layer**, where we include Plotly for data visualization, which integrates really beautifully with Streamlit and allows to have more interactive and responding visualizations than other similar libraries. On this layer we also find pandas, the essential library for dealing with DataFrames (tabular data) and NLTK and OpenCV for working with text and visual content in that order.

One of the key layers of the stack is the **AI/ML Models Layer**, which includes on the one hand ML frameworks for building custom ML models from scratch or customizing existing ones (TensorFlow, PyTorch and HuggingFace) and on the other the main cloud vendors that provide high-level AI solutions in the form of APIs.

Finally, the other key layer in our Tech Stack is the Web Application Layer, which leverages the capabilities that the previously introduced Streamlit has to offer for rapidly building data apps and FastAPI, a web development framework to easily expose as independent endpoints the custom ML models.

Let’s go back to imagining that we are an insurance company that wants to define a potential prototype for this sector. We have identified a problematic that consists of manually processing the thousands of emails that we receive from customers regarding car damage claims. These contain explanations on what happened to their vehicle together with personal information (e.g., personal addresses, driver ID, email) and some images of the damages. We would like to use AI to identify all the personal information in the text and classify the type of damage from the image.

We consider that before spending lots of efforts and resources into developing a large solution, it would be worth prototyping this use-case in order to validate its technical feasibility and its business impact by sharing it with the C-level executives.

One of the key layers of the stack is the **Web Application Layer**, where we include Plotly for data visualization, which integrates really beautifully with Streamlit and allows to have more interactive and responding visualizations than other similar libraries. On this layer we also find pandas, the essential library for dealing with DataFrames (tabular data) and NLTK and OpenCV for working with text and visual content in that order.

One of the key layers of the stack is the **AI/ML Models Layer**, which includes on the one hand ML frameworks for building custom ML models from scratch or customizing existing ones (TensorFlow, PyTorch and HuggingFace) and on the other the main cloud vendors that provide high-level AI solutions in the form of APIs.

Finally, the other key layer in our Tech Stack is the Web Application Layer, which leverages the capabilities that the previously introduced Streamlit has to offer for rapidly building data apps and FastAPI, a web development framework to easily expose as independent endpoints the custom ML models.
the other hand, we want to know the type of damage that the vehicle in the image has. For this, we will need to train a custom model. We could base our computer vision model on a ResNet50, fine-tune it on the Peltarion Car Damage Dataset [5] using PyTorch, and easily wrap it inside a REST API using FastAPI.

Once we have our AI models ready, we can turn our Python scripts and Jupyter Notebooks into something more interactive, shareable, and easier to present using Streamlit with little effort. We realize that we have validated the technological viability of this use-case and have something tangible that we can show to our stakeholders to refine and validate our prototype iteratively until we are satisfied with the result.

In the previous section, we showed how we can rapidly materialize our ideas into prototypes using the AI Technology Stack and the Prototyping Studio. However, these prototypes could be easily scaled up thanks to our AI Labs and AI Driven Design methodology. The former acts as an innovation center, facilitating rapid experimentation with prototypes and serving as the first step towards the creation of Intelligent Digital Services.

The latter merges business strategy with the design and development of solutions to identify, test, and scale Intelligent Digital Services. This is achieved thanks to the integration of multidisciplinary teams where specialized profiles collaborate in the different phases of the solution combining business expertise with the highest technical excellence. Allowing our customers to create value and keep accelerating their innovation process across the different stages of the service conceptualization, design, development, and deployment.
CONCLUSIONS

Speeding up innovation and unlocking the full potential of AI initiatives has become a real puzzle for companies across sectors, mainly because of the lack of environment and tools to rapidly create and easily manage the fast production of AI prototypes. At NTT DATA, we have been working on multiple AI projects and experimented with different AI capabilities and tools, letting us define an ever evolving and Python centric technology stack that increases the speed of experimentation. Prototyping let us refine and validate use cases iteratively, reducing the final costs and achieving a high-quality product in a reduced amount of time. Lastly, prototypes can be easily scaled up into Intelligent Digital Services thanks to our AI Driven Design methodology creating high-value services for our customers.

References
1. “Why do 87% of data science projects never make it into production?”, Venture Beat


3. Crunchbase Streamlit

4. Streamlit GitHub

5. “Car Damage Assessment”, Peltarion
CONTACT US FOR FURTHER INFORMATION ABOUT THIS PRACTICE:

DAVID PEREIRA PAZ  
Head of Data & Intelligence Europe

JACINTO ESTRECHA  
Head of Artificial Intelligence Spain

ADIL MOUJAHID  
Technical Manager  
@ AI Center of Excellence

ANDREA CORNAVACA  
Lead Analyst AI Strategy  
@ AI Center of Excellence

MARCH 2022  
© 2022 NTT DATA, Inc. All rights reserved.