Åbo Akademi University

EDISS MASTER'S PROGRAMME

Internship Report

Ericsson AB, Stockholm, Sweden

Submitted by: Saad Waseem

Student ID: 2201865

Duration: 29.05 - 01.09

2023

Table of Contents

The Company

Ericsson AB | HQ | Stockholm, Sweden

Ericsson AB is a multinational telecommunications and networking company headquartered in Stockholm, Sweden. Ericsson's organizational structure typically includes various business units and divisions focused on different aspects of telecommunications and technology. They are known for their Networks, Digital Services, Managed Services, and Emerging Business segments. Within these segments, the company operates teams (also known as Group Functions) specializing in research and development, sales, marketing, and customer support.

https://www.ericsson.com/4a46fa/assets/local/about-ericsson/corporate-governance/images/ericsson-organization-chart-july-21-2022-jpeg.jpg

Mission/Objectives:

Ericsson's mission revolves around enabling communication for people, businesses, and societies. Their core objectives include:

- Connecting the World: Ericsson aims to provide advanced communication technologies and infrastructure to connect people globally.
- Empowering Digital Transformation: They work to empower businesses and governments with the tools and networks necessary to drive digital transformation.
- Sustainability: Ericsson is committed to sustainability and strives to create innovative solutions to reduce their carbon footprint and enable sustainable development.

Present and Future:

Ericsson, since its inception in 1876, is known for innovation and developing cutting edge solutions for telecommunication and advanced networks. Currently, they are heavily involved in the deployment of 5G networks and the development of the Internet of Things (IoT) ecosystem. Ericsson's Research department is actively engaged in the development of novel techniques and methods to support 5G, artificial intelligence, and automation. As a responsible organization Ericsson is committed to work towards Net Zero and reduced carbon emission by 2040 (Ericsson).

Ericsson Research | Trustworthy AI | TWAI

Ericsson Research department has multiple teams working in different areas of advanced research in artificial intelligence, machine learning, 5G networks AR/VR etc. I was working with their Trustworthy AI team lead by Rafia Inam who was also my manager during the internship. TWAI team is mostly working on devising techniques and solutions for integrating explainable artificial intelligence (XAI) in different domains of telecommunication whether it is 5G network slicing, automation and other related areas like compliance.

Tasks and Responsibilities

Right after the onboarding, I had a few meetings with my mentors from TWAI team who explained to me the project scope, my primary responsibilities and concepts that I should understand before diving into proof of concept development.

Research

Background - Domain: The TWAI team is working on developing different techniques to integrate explainability within various machine learning pipelines specifically applicable to telecom use cases. The main motivation behind these efforts is to explain predictions made by a selected machine learning model consequently making it understandable for stakeholders from business and legal teams.

Literature Review: Initially, I was provided with a list of papers that I should go through along with a pre-recorded conference from Stanford. The focus of the initial two weeks was to help me understand different explanation methods that can be used with different modalities. This effort was not only limited to reading papers but also involved experimentation so that I get hands-on experience working with different libraries like LIME (Local Interpretable Model-agnostic Explanations), SHAP (Shapley additive explanations), eli5 (Explain Like I'm 5), Captum and be able to understand what does it actually mean when we talk about explainability in the context of AI. References section provides a list of research papers that I reviewed to enhance my understanding of XAI before diving into the specific use case that I would be working on in the following weeks.

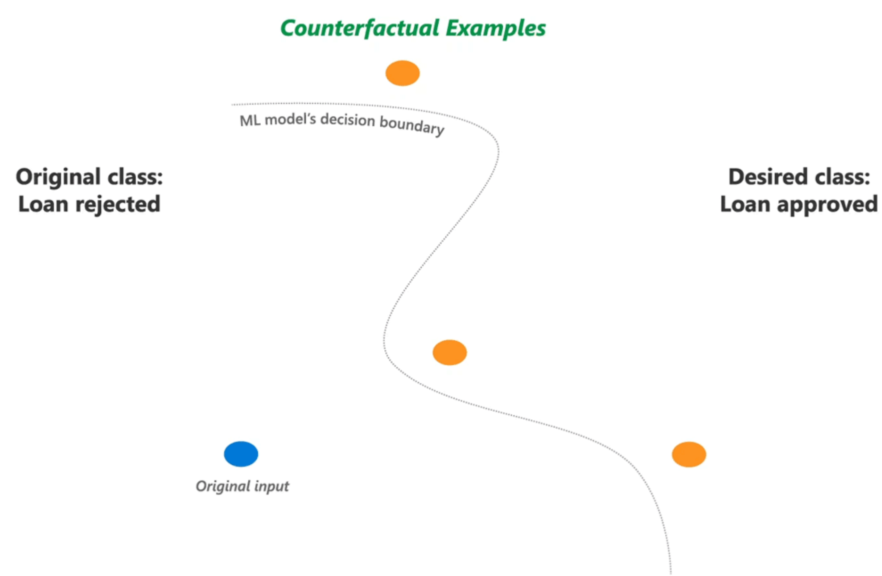
POC's Use Case

Since Ericsson is a telecom company their research efforts vastly surround emerging network technologies. For the same reason, POC's use case was focussed on selecting explainers based on specific metrics using their published research about SLA Violation Prediction in 5G Network. There are various different explainers available that can be used with different kind of machine learning models however, our focus was on the following ML models and their corresponding explanation method:

Model	Explainer	Dataset (Tabular)
Logistic Regression	Counterfactual (DICE & Wachter)	IRIS, Diabetes, Telecom
Random Forest	Counterfactual (DICE & Wachter)	IRIS, Diabetes, Telecom
Decision Trees	Counterfactual (DICE & Wachter)	IRIS, Diabetes, Telecom
CNN (PyTorch)	Local Attributive Explainers (IG, DL, LRP), Global Attributive Explainers (IG, DL, GS, DLS)	Telecom

POC supported Explainers

Counterfactuals base their approach on 'what if' scenarios. In simple terms, a counterfactual is defined as the smallest change in input features that can achieve the desired state. "For example, consider a person who applied for a loan and was rejected by the loan distribution algorithm of a financial company. Typically, the company may provide an explanation on why the loan was rejected, for example, due to "poor credit history". However, such an explanation does not help the person decide what they should do next to improve their chances of being approved in the future. Critically, the most important feature may not be enough to flip the decision of the algorithm, and in practice, may not even be changeable such as gender and race." (DiCE)



Counterfactual Explanation

I incorporated two counterfactual implementations (DiCE & Wachter) to let users experiment with different counterfactuals based on varying user preferences like proximity and diversity etc, to ensure validity and applicability of the generated counterfactuals.

Attributive Methods provide explanation in terms of features importance scores. They can use different mathematical approaches like gradients and activations to calculate the importance. Moreover, these methods behave differently depending upon how and how many baseline instances are being used that make them local or global in their approach. The challenge was to draw comparisons between different explanations for analysis and to calculate certain metrics (that Ericsson researcher has developed) to indicate effectiveness of explanation methods.

I was responsible for incorporating these metrics in POC implementation and developing user controls so that experiments can be conducted using varied configuration in order to understand different explanations across machine learning models and datasets.

Development

Technology Stack

For the development of POC we chose Python 3 as a programming language, Anaconda for python environment management, Streamlit as a tool for development, VS code as an editor for programming. Git & Gitlab was used for source version control.

Deployment Stack

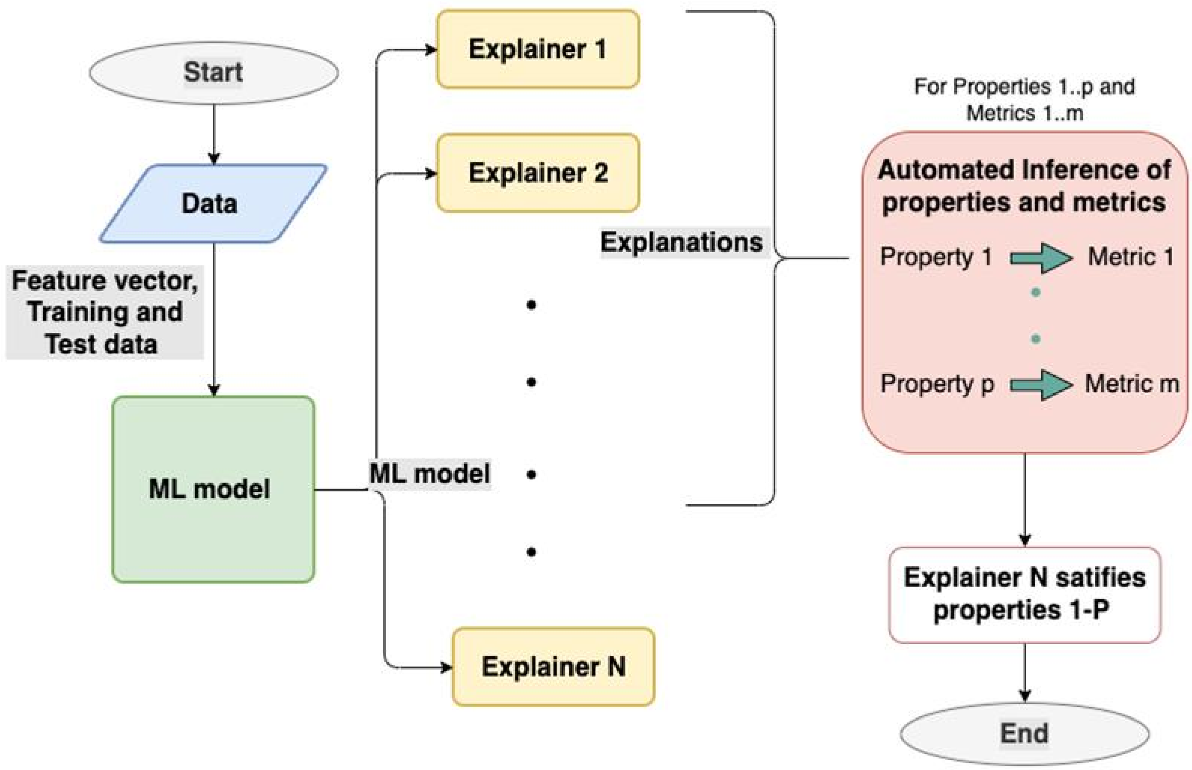
For deployment, Ericsson follows a containerized environment using Docker containers with Kubernetes (K8s).

Development Tasks

Here is the list of development tasks that I performed during my internship:

1. **Setup Development Environment**
   As with any development project I set up the local environment with required dependencies. There were a few gitlab repos that contained basic implementation of Counterfactuals and local attribution methods with their corresponding metrics previously developed by the team. I reviewed the implementation and figured out the approach for incorporating existing code in streamlit for POC development.
2. **User interface (UI) and Streamlit**
   The reason I suggested using streamlit for POC development is because it offers a lot more flexibility in terms of UI development. Initial requirements for the POC are fully aligned with many out of the box features that streamlit supports. On top of it, I applied customization using Streamlit APIs to meet changing requirements that followed as I made progress with the development.
3. **Core Functionality**
   I implemented the complete pipeline which was divided into 3 steps:
   a. **ML Model Training**
   b. **Generate Explanations**
   c. **Quantify Explanations**

ML Model Training (Step 1): The user can choose a combination of machine learning model and a dataset as pictured in Table 1. The selected model is then trained on the data before explanations can be created. The choice of type of explanations is also configured by the user as a first step and UI dynamically adjusts controls to meet specific explanation requirements. Another possibility for ML model was to let the user upload a compatible pair of dataset and pre-trained model in that case the training section would be skipped and now the user can head towards explanation generation.



POC Pipeline

V.Singh, | K.Cyras, R.Inam, “Explainability Metrics for Counterfactual Explanations" EXTRAMAAS 2022 [Link], FULLTEXT01.pdf

Generate Explanations: Based on user selection in step 1, the user can either generate counterfactuals or attributions. For counterfactuals, I incorporated two approaches one was Diverse Counterfactual Explanations (DiCE) the other approach was Wachter’s method. It was a crucial requirement for the POC to let the user play with different preferences like proximity, validity, diversity etc for counterfactuals and few other metrics related to attributive methods. As a final step the user could choose any instance from the test dataset for explanation. The generated explanation is intuitively presented on-screen with weightage assigned to configured user preferences for further analysis.

Quantify Explanations: Quantification of explanations is a unique feature of this POC as it makes it possible for the user to compare pairs of explanations in terms of certain metric scores developed by Ericsson Researchers. These results are then visually presented as graphs for an easy comparison and meaningful explanations. I am omitting a few details here as this work is not yet published and I am advised to hold off this information until then.

4. **Performance**
In order to keep the solution performance efficient I followed Streamlit coding standards and also implemented caching to avoid unnecessary recomputation where the objects can be reused.

5. **Status Update | Progress**
For communication and status update meetings we were using Microsoft Teams. Apart from day-to-day communication, I used to have weekly meetings with my supervisors, one researcher from Ericsson Sweden and the other from Ericsson USA in retrospect to take their feedback and plan for new features.

6. **Final Presentation | Demo**
The efforts done during the three months internship resulted in a fully functional POC that could be hosted locally by anyone at Ericsson using system wide python installation or using docker containers by pulling the docker image from Ericsson's hosted docker registry. I gave my final presentation and showcased the features of the POC to 40+ participants from different departments of Ericsson followed by an interactive session of Q & A.

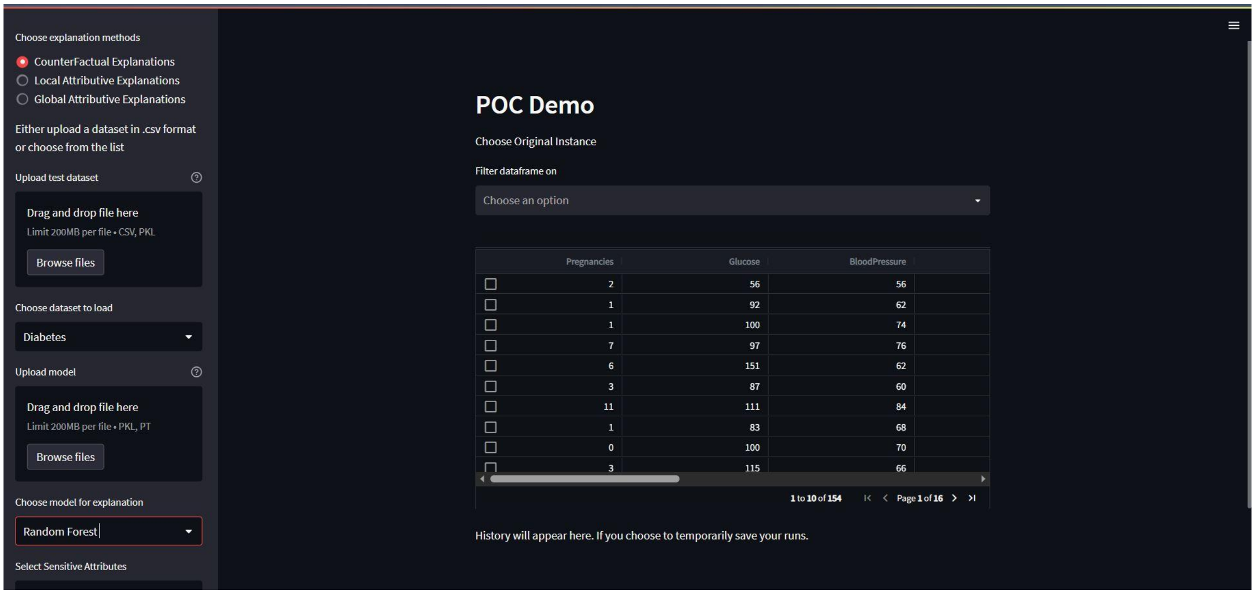


Figure 1: Ericsson proof-of-concept for three types of explanations CFs, LAE, GAE
CFs: counterfactuals, LAE: local attribution, GAE: Global attribution

Learning Outcome

The learning outcomes can be splitted in two heads namely research and development:

Research Outcome

The internship provided me with an opportunity to explore most recent research in the field of *Explainable Artificial Intelligence* and to understand the importance of explainability for enterprises like Ericsson. I also learned about industrial research and how domain experts play their role in validating the outcomes of research efforts within a specific domain.

Development Outcome

I learned about Anaconda for python development that is typically used for virtual environment management to segregate project specific packages. I also learned about Streamlit which is an open source tool supported by a huge community of researchers and developers for POC development and conducting quicker experiments with data science and machine learning projects. Additionally, I extensively worked with a few python libraries commonly used for explanations like ML Extend, DiCE (Microsoft), Captum (Facebook) etc.

Future Prospects

I am pleased to share that my previous experience with development projects and the most recent courses like Data Science, Machine Learning and Data Intensive Engineering that I took during my first year in EDISS helped me perform with confidence, learn new concepts and meet the expectations of my employer. They all expressed their gratitude with my performance during the internship and were satisfied with the delivery of POC. They also expressed their interest in working with me for two potential threads: one is a possibility to publish a *Tool Paper* in an international conference and the other could be a potential *Thesis Project* that they might offer towards the end of this year (2023).

References

Research Papers

1. *Saranya A., Subhashini R., A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends* [DOI]
2. *Ahmad Terra, Rafia Inam, et. al, Using Counterfactuals to Proactively Solve Service Level Agreement Violations in 5G Networks* [DOI]
3. *Ahmad Terra, Rafia Inam, et. al, Explainability Methods for Identifying Root-Cause of SLA Violation Prediction in 5G Network* [DOI]
4. *Sandra Wachter, Brent Mittelstadt, et. al, Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR* [DOI]
5. *Ramaravind Kommiya Mothilal, Amit Sharma, et. al, Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations* [DOI]
6. *Vandita Singh, Kristijonas Cyras and Rafia Inam, Explainability Metrics and Properties for Counterfactual Explanation Methods, 4th International Workshop on EXplainable and TRAnsparent AI and Multi-Agent Systems (EXTRAAMAS 2022)* [DOI]
7. *LRP based explanation tool* Explainable AI Demo
8. *Timo Speith, A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods* [DOI]
9. *Guidotti, R. Counterfactual explanations and how to find them: literature review and benchmarking. Data Min Knowl Disc (2022)* [Link]

Online Workshop

Professor Hima Lakkaraju, Machine Learning Explainability Workshop, Stanford

Technical Documentation

1. Diverse Counterfactual Explanations (DiCE)
2. Github DiCE ML
3. ML Extend
4. Explain like I'm 5 (eli5)
5. Captum - Model Interpretability for pyTorch
6. Python Anaconda - Environment Management
7. Streamlit Build Faster