INTRO

Hello, my name is Helen and I'm a Data Scientist and Machine Learning Engineer with over 2 years of experience, specializing in both technical and business domains. My background in mathematics and programming has equipped me with understanding of the data science workflow, from addressing business problems to deploying machine learning models in production environments.

With a sharp attention to detail, I excel in leveraging statistical methods, machine learning techniques, and data analysis to extract insights from complex datasets. My passion lies in exploring various business domains and devising innovative strategies to apply machine learning models effectively. In terms of technology stack, my initial education and specialization was in the field of classical machine learning, which allows them to be well versed in many classical algorithms, such as linear regression, logistic regression, decision trees, random forests, kNN, SVM, k-means clustering, and dimensionality reduction algorithms like t-SNE and PCA, ensemble models (Ensemble models: Bagging, Gradient Boosting, Stacking, and also time series forecasting models like SARIMA, Prophet, LSTM.

But as time goes by my expertise in machine learning spans across a variety of DL frameworks including TensorFlow, Keras and PyTorch. Additionally, I have hands-on experience with cloud services such as AWS and Azure, utilizing a range of tools like AWS Lambda, SageMaker, Azure ML, and more for model deployment and management.

Furthermore, I am well-versed in utilizing databases like PostgreSQL and Redis, as well as MLOps tools including Docker for seamless deployment and scalability. My proficiency extends to source control systems such as Gitlub, GitHub and Bitbucket for efficient collaboration and version control.

In short, my diverse skills and experience allow me to create and implement machine learning models, smoothly incorporating them into business operations across different fields. I'm eager to explore how I can support your team's achievements with my expertise.

PROJECT 1

During my time at Innowise Group, I worked on several projects, the first one was the Sales Prediction Engine. The task of this project was to build an analytical engine for monthly sales volume forecasting using advanced machine learning algorithms. This engine had to be trained each time new sales volume data became available in order to prevent data drift.

My responsibilities and achievements on the Sales Prediction Engine project included:

1. Conducting in-depth exploratory data analysis
2. Developing a statistical outlier detection layer
3. Training and testing multiple time series approaches, such as the Cat Boost, ARIMA, Facebook Prophet, and gradient boosting decision trees (GBDT) algorithms.
4. Developing a robust model that can adapt to monthly changes due to a thoughtful approach to generating new features
5. Providing valuable analytical reports to the end user and proposing and integrating a recommendation system that offered relevant advice to businesses

Additionally, I utilized various tools and techniques:

- Used dvc for version control of data, which was updated monthly in the S3 storage.
- Implemented a CI/CD process in Git Actions to automate model deployment and updates.
- Utilized mlflow for experimentation and model tracking.
- Set up a pipeline with Airflow for data loading, preprocessing, and model training on new data to prevent data drift, considering the emergence of new items and stores.
- Developed a robust model capable of handling changes in the list of stores and products every month due to a thoughtful approach to generating new features.

The final product would consist of two main components:

1. CSV Exports:
   - Predicted sales data in CSV format for the specified period, providing a detailed breakdown of sales forecasts for each product and shop.
2. Tableau Dashboard:

An automated dashboard in Tableau, offering intuitive and interactive visualizations of sales predictions. It features various charts, graphs, and tables. In summary, the final deliverables would provide both structured data exports in CSV format for further analysis and an interactive visualization tool in the form of a Tableau dashboard, enabling stakeholders to gain valuable insights and effectively utilize the sales forecasts for strategic planning.

Overall, working on this project provided a comprehensive understanding of sales forecasting methods and further developed skills in data analysis, model development, and presenting ideas to stakeholders.

Difficulties that we encountered in the process of developing and implementing this project:

The biggest challenge in developing such a project was that the developed model had to adequately predict data that was initially missing in the training dataset, for example, it would be necessary to predict how certain products could be sold in a new store opening. Here it was necessary to carefully work on the generation of signs and make sure that there was no data leakage

PROJECT 2. Software for legal companies

The second project I participated in was a software solution for legal companies. It's an application designed for employees of large law firms to streamline the drafting of standard legal contracts. It utilizes AI to analyze existing company documents and employs an algorithm to segment documents into reusable logical parts.

Process of the final product:

Process of the final product: The user uploads a document into the Word editor and activates the plugin. The plugin analyzes the document, extracts key phrases, and structures the text. Recommendations for correcting the text and improving the structure are displayed in the plugin's

user interface. The user can review the recommendations, make changes, and save the processed document directly in the Word editor.

Additionally, the user can input a request for a contract template by specifying keywords (such as company and individual names, contract subject, important terms), and the plugin suggests the most relevant contract from the existing database, of course, pre-cleansed of personal information using Azure services.

Here I must say that the entire project from start to finish was implemented using Microsoft Azure services such as Azure Cognitive Services, Azure Machine Learning, Azure Blob Storage and Azure SQL Database. The customer chose this option; it seemed more reliable to him. But we also proposed another implementation of this project, I can tell you about it later, if you want.

The general plan we followed when implementing this project was as follows:

1. **Requirement Gathering:** Understanding the needs and challenges faced by employees of large law firms in drafting standard legal contracts efficiently.
2. **Conceptualization:** Brainstorming and conceptualizing the idea of using AI to streamline the drafting process by analyzing existing company documents and segmenting them into reusable logical parts.
3. **Selection of Technology Stack:** Choosing Microsoft Azure services such as Azure Cognitive Services, Azure Machine Learning, Azure Blob Storage, and Azure SQL Database for their reliability and suitability to the project requirements.
4. **Designing the Architecture:** Creating a plan for the project architecture, outlining how each Azure service would be utilized to fulfill specific tasks such as text processing, document analysis, user interface development, data storage, and security.
5. **Development of Text Processing and Recognition Module:** Implementing Azure Cognitive Services such as Computer Vision for text recognition in document images, extracting OCR results, and utilizing Form Recognizer for structured data extraction from various document types.
6. **Text Analysis and NLP Implementation:** Leveraging Azure Cognitive Services Text Analytics for text analysis, including key phrase identification, named entity recognition, and personal information detection. Ensuring documents are cleansed of personal information to uphold confidentiality rights.
7. **Data Labeling and Model Training:** Utilizing Azure Machine Learning for model training on provided documents, including data labeling to enhance accuracy in recognizing and analyzing new documents.
8. **User Interface Development and Integration with Microsoft Word:** Developing a plugin for Microsoft Word that provides access to document analysis functionality. This plugin will activate based on user access. It will include document upload, processing, analysis, recommendations, and corrections directly within the Word editor interface.
9. **Data Storage and Management Setup:** Implementing Azure Blob Storage for storing raw and processed documents, and Azure SQL Database for managing structured data and analysis results.
10. **Data Security and Confidentiality Implementation:** Utilizing Azure Active Directory and Azure Key Vault for user authentication and data access control. Azure Key Vault manages keys and secrets essential for secure data access.

11. **Scalability and Performance:** The architecture will be designed to scale horizontally and vertically to handle increasing document processing loads efficiently.

12. **Monitoring and Logging:** Azure Monitor and Azure Log Analytics will be used for monitoring the plugin's performance, identifying issues, and logging relevant data for analysis. Since the solution is based on a Microsoft Word the focus of monitoring is on maintaining and updating the plugin to ensure its smooth operation within the Word environment.

Difficulties that we encountered in the process of developing and implementing this project:

1. The data that was stored in the company's contract database was confidential and in order to reuse it, it was necessary to depersonalize it. To automate this process, we used entity recognition from Azure Cognitive Services (Text Analytics).

2. The second rather complex and unusual task was the requirement of checking the contract for the completeness of the structure. To solve this problem, we used Text Analytics to extract key phrases and entities from the contract text, which are compared against a predefined list of mandatory blocks to ensure that all necessary sections are included and thus we were able to organize the process of highlighting any missing or incomplete sections.

Overall, the process of implementing this project was highly engaging. It required a deep dive into the complexities of drafting legal documents to effectively oversee, correct, and enhance the functionality of the final product.

PROJECT 3. Software for agricultural robots

My third project that I have been working on for the last six months and until now is Software for agricultural robots. Research and development of an AI-enabled software for agricultural robots for both Pre-harvesting stage (crops and weeds detection, disease spotting, field state recognition with RGB and NIR images) and Harvesting stage (fruit number and maturity level estimation using Sonar technology data).

These technologies offer a comprehensive solution to modern agricultural issues by automating tasks and optimizing resource usage.

Project customer produces autonomous robots for cultivating and nurturing plants. Initially, the robots could move around fields but lacked the ability to differentiate between plants and weeds for selective fertilization and watering. Our experts integrated software into the robots to distinguish and segregate plants accurately. The program's goal was to eliminate specific weeds using lasers with optimal accuracy and supply plants with suitable fertilizer based on their class and condition metrics.

We implemented an AI solution for real-time processing of scanned field images, enabling the robots to identify weeds in milliseconds. Equipped with calibrated lasers, the robots can eliminate

up to 100,000 weeds per hour and classify plants, administering fertilizers based on their requirements. They can also determine field conditions and metrics to optimize agricultural practices.

To achieve this, we gathered and labeled a dataset of over 10,000 plant images using an integrated video camera. Tasks such as marking, augmentation, and model training were performed on the dataset. We implemented a supervised machine learning model that can predict stem and field images, enabling further plant classification, stem detection, weed eradication, and selective fertilization.

Additionally, we developed a custom neural network for plant identification and treatment decision-making, integrated into an end device with GPU for real-time processing. The software allows the robot to make decisions without internet access, updating the dataset when connected to the network. The neural network can be retrained using updated datasets to accommodate new plant types and weed types.

Our plan for implementing the software for agricultural robots was following:

1. **Requirement Analysis:**
   - Understand the specific challenges faced in agricultural processes.
   - Identify the need for tasks like crop and weed detection, disease spotting, and field state recognition.
   - Determine the requirements for harvesting tasks like fruit number and maturity level estimation.
2. **Conceptualization:**
   - Brainstorm ideas for developing AI-enabled software to address the identified needs.
   - Consider leveraging technologies such as machine learning and computer vision to achieve the desired functionalities.
3. **Development of Data Processing and Model Training Pipelines:**
   - Implement infrastructure for data processing and model training.
   - Consider both research and production workflows.
   - Utilize PyTorch for research applications and Docker for production deployment on Kubernetes (K8s) clusters hosted in AWS.
4. **Experimentation with Model Architectures:**
   - Experiment with different architectures like UNet, DeepLab, Mask R-CNN, etc., for crops and weeds segmentation.
   - Adapt these architectures for RGB+NIR channels to automate pesticide application and plant feeding.
5. **Training and Fine-Tuning Models:**
   - Train networks with various architectures for different use cases such as field state classification, crop disease spotting, fruit number, and maturity level estimation.
   - Fine-tune models based on specific tasks using transferred backbones.
6. **Model Deployment to Edge Devices:**
   - Deploy trained models to edge computing devices for high-speed inference.
   - Focus on efficiency in the field.
   - Utilize quantization techniques to optimize model inference performance.
7. **Deployment Workflow:**

- Implement a deployment workflow using Jenkins and Artifactory for production deployment.
- Ensure seamless deployment of models to remote machines in the field.

8. Monitoring and Evaluation:

- Use platforms like Weights & Biases for monitoring and evaluating machine learning runs.
- Integrate W&B logging into the codebase to monitor training runs, training and validation loss.

9. Presentation of Results:

- Regularly present model speed and robustness performance improvements to both internal teams and external stakeholders.
- Highlight the benefits and impact of the software on agricultural processes.

Overall, the ML farm system developed by our ML team showcases the benefits of machine learning in agriculture, providing cost-effective and efficient solutions for crop management and treatment.

The ML farm systems use high-precision lasers for weed elimination and selective plant feeding, providing accurate semantic segmentation of crops, weeds, and grass. This allows the autonomous weeders to kill over 100,000 weeds per hour without chemicals. The selective feeding system analyzes each plant's state and applies the most appropriate treatment, reducing resources and cost.

In the Software for Agricultural Robots project, documentation was a crucial aspect of our work. We maintained detailed documentation to ensure clarity and consistency across the team. This included documentation for project requirements, design decisions, architecture, and implementation details. We used tools like Confluence and Google Docs to create and manage our documentation, making it accessible to all team members.

As a Researcher / Machine Learning engineer in the team, we followed an agile methodology, specifically Scrum, to manage our project. Our team consisted of a researcher, two machine learning engineers, a software developer, and 1 domain expert. We held regular scrum meetings, including daily stand-ups, sprint planning, estimation of tasks, grooming sessions, and demos.

During daily stand-ups, each team member would share their progress, any blockers they were facing, and their plans for the day. Sprint planning meetings were held at the beginning of each sprint to determine the goals and tasks for the sprint. Estimation of tasks was done using story points to estimate the effort required for each task.

Grooming sessions were used to refine the backlog and ensure that the team had a clear understanding of upcoming tasks. Demos were held at the end of each sprint to showcase the completed work to stakeholders and gather feedback.

Overall, the agile approach helped us stay organized, collaborate effectively, and deliver high-quality software for agricultural robots.

Difficulties that we encountered in the process of developing and implementing this project:

1. The first problem was training the neural network model that identifies crops and weeds. This is a challenging problem because many weeds look just like crops. Professional agronomists and weed scientists trained our labeling workforce to label the images correctly.

2. If the robot needs to drive more slowly to wait for inferences, it can't be as efficient in the fields. So one of our top priorities is high-speed inference on the edge computing device. Since all our inference happens in real time, uploading to the cloud would take too long. We Appled quantization technique to boost model inference performance and meet hardware restriction and integrated this solution into an end device equipped with GPU, allowing it to process real-time data and distinguish plants from previously learned datasets.