Front End

**Requirements**

In order to have a final user being able to click a button to run the code and see the results, an application needed to be created and deployed to a server, so it would be available online and could communicate in real-time with the backend service.

The following requirements were defined prior to the application development:

1. The interface would contain tabs for the user to navigate and see different visualizations and metrics. The available tabs are the following:
   1. Model Training
   2. Dashboard
   3. EDA
   4. Resources
   5. Group
2. An end user with no previous knowledge would be able to run the application and see the results using the Model Training tab.
   1. The results for the machine learning process (model accuracy, best model, etc), would be available in a table format for the user to see and compare.
   2. A weather forecast for the next 7 days (including days and nights) would be available in a table format with the package volume forecast predicted using the chosen machine learning model (the application automatically picks the best model to use)
3. A Power BI dashboard containing the historical data information and the forecast for both weather and package volume would be available under the Dashboard tab. Some interactions such as filtering data by city and period of time is also a feature that users may see.
4. A complete EDA (Exploratory Data Analysis) would be available under the EDA tab, containing a generated HTML file rendered by the interface with a complete analysis of all the features used in the machine learning process ()

**Scope**

**Approach**

Back End

**Requirements**

A backend service written in python and using data science concepts to do transformations and run a machine learning process was created in a way it would be available online as a Rest API, waiting for the user to hit an endpoint to run the process and show the results in a JSON format.

**Scope**

**Approach**

LESSONS LEARNED

[How to write a lessons learned report as a project manager | Indeed.com UK](https://uk.indeed.com/career-advice/career-development/lessons-learned-report)

(What did you wish you knew before the project started!, that would have made your work much easier, or would make you avoid mistakes and waste of time, what went perfectly and you would definitely repeat also)

Project Management

The tools, and environment for collaboration and planning could have been defined in the first week, and presented more clearly to the team in a meeting designed for that, this would have enhanced a sense of organisation and goals, and would have allowed for more proactiveness and suggestions on the starting period.

Meeting always at mondays to see where everyone is at with their work, discuss and make everyone get “the picture” of our project was of great benefit, facilitating collaboration, permitting suggestions, support, and proactivity, most of those meetings were quite productive, and even the two that were less productive worked to bring a sense of security to the team that would make us know that we have our project under control and wouldn't deliver late.

Adapting Sprints using a week(the common practice is using a day), worked perfectly because the time we were allocating weekley, 5-8 hours, matched that of a Day of Work.

Defining a persona for our target, the person who would in theory use the interface to make decisions, in our case an operations manager inside a Canada Post facility, helped us better focus our efforts and define and limit our Scope. If this process wasn't done, and talked to the sponsor, we risked spending resources on a direction that would later be scrapped, so sending an email to the Sponsor before continuation of work on those areas was a wise decision.

Data Mining

Data mining in weather information can be challenging due to the complexity and sheer volume of data involved. We got over 50 attributes in the historical data table. Weather data, especially the forecast data is typically collected at a high frequency, with multiple measurements taken every hour. This results in large datasets that can be difficult to process and analyze. So in our case, we used daily data from weatherstats.ca which is a mirrored site to Environment Canada which guarantees the accuracy of the data..

Another challenge is the variability and unpredictability of weather patterns. Weather data can exhibit complex and non-linear relationships, making it difficult to identify patterns and correlations. It is important to carefully preprocess and clean the data to remove outliers and errors that may affect the accuracy of the analysis.

Data Transformation

The data transformation step can get really complex depending on the data and the validations required. Using python to handle this process was a good way to achieve good results and end up with a reusable script, as there are many built-in functions and libraries that make working with data an easier task. Even automating data extraction and merging dataframes created from the raw data extracted was possible and relatively scalable as we were reusing the working code..

Data Preparation

In order to get the most optimal and accurate prediction, the majority of the preparation was tailored towards regression data preparation. All the steps taken from dropping columns that have a missing value greater than 60%, replacing the rest with the mean value, label encoding the categorical features, expansion and preparation of the calendar date column.

Machine Learning

For the purpose of exhausting all option in order to get the highest predictive model, we applied the data to multiple models, each model has a set of hyper parameters in which we use Bayesian Search CV to iterate through all the possible combinations of the hyperparameters for each model in order to get the best performing one. But this was done with limitations of the number of iterations to be performed and the search space (number of hyper parameters) to be applied in order to achieve the best model, reason for this was run time as the larger the search space and number of iterations the longer the processing time.

Cloud & UI - Vinicius

For practical availability we chose to move the application to Microsoft Azure. Our application consists of a python backend Rest API and having it to run online was a good challenge. The frontend was created in ReactJS and hosted by github pages. Both applications were controlled by github actions for the automated deployments, so everytime we needed to update the code for either application, a simple “push” to github would handle the whole deployment process through github “actions”.

Once both applications were running, we just needed a user interaction to run the ETL and machine learning process related to the Volume forecast and weather forecast for the next few days.

Visualizations

One of the issues faced was the unreliability of data format from weather prediction sources. Temperature values sometimes returned a string, explaining weather behaviour instead of temperature integers. Trying to display those fields as a bar or line chart is prone to errors.

Due to limitations of our Power BI plan, some processes couldn’t be satisfactorily automated. There are also limitations regarding creating dedicated workspaces, and also users that can access it online.

It’s also important to highlight that small variance in key column values could significantly harm relationships and the report ability to filter data conveniently in the dashboards. Power Query fixed some of the issues that weren’t dealt with in the ETL steps.

Given the amount of 0 values in the dataset, some DAX measures were used to calculate Average and median values disregarding 0 values. These measures might be more useful for KPI metrics.

NEXT STEPS

(What would you do to deliver a better result or/and if you had more resources, and what limitations were an obstacle to do more)

Project Management

To put the Model, and UI in motion, we would need to change from a test concept version to production version where we would need to aggregate more data and processing, to have a model that would be accurate and usable in real time. A weekly cycle would be moved to a daily Sprint like the usual application, therefore we would speed up the testing and improvements, making also that the results are more usable for decision making.

After this production cycle, a testing phase would occur with either one location or a simulation during a period of at least one month, where final adjustments and reliability could be checked for a later implementation on smaller facilities where risk of eventual problems could be reduced and better managed, while also providing insights that would allow to enhance even further the forecasting.

Data Mining

Use API to get near real time weather forecasts from <https://api.weather.gc.ca>. Also, experimenting with the GeoMet-OGC-API, it provides public access to the Meteorological Service of Canada (MSC) and Environment and Climate Change Canada (ECCC) data via interoperable web services and application programming interfaces (API). We can freely and quickly access thousands of real-time and archived weather, climate and water datasets. We can build mobile apps, create interactive web maps.

Data Transformation

For the data transformation, doing some asserts inside the code would guarantee the data integrity over the time.

Data Preparation

The next steps that can be applied to the data preparation would be to create separate dataframes, first one being the originally prepared data, second one being the original prepped data but normalized and the third being the prepped data but standardized, in order to test the accuracy of each of the three to increase the chances of acquiring a higher accuracy.

Machine Learning

Next steps for this should be to widen the search space and introduce a larger set of hyperparameters in order to exhaust all possible combinations to find the best performing model. One other thing that can be further implemented is the use of Ensemble Learning, the models that were applied to the data are individual models, if we apply ensemble modeling then maybe it will outperform the best chosen model out of the individual models. The final next step that can be applied is to calculate feature importance, this allows for the system to detect which features have the least impact to the accuracy and decide whether to drop or not and test to see if there is an increase in accuracy.

Cloud & UI - Vinicius

A suggestion on the application part would be to create unit testing on the backend code to assure the code is working as expected and the calculations are all working. Creating a staging environment only for the users to test would also be a good suggestion, together with the automation for the staging and production environments.

Visualizations

Changing ownership of the Power BI file could facilitate access to Canada Post Users, especially when using a premium version of this tool. Another suggestion would be testing if using Power BI web application instead of only using the UI we created could improve the Data democratization process.