Evaluation:

The workflow developed in KNIME Analytics Platform aimed to identify the best model for predicting total duration or ETA in the dataset. Three different regression models were implemented and evaluated – Random Forest, Simple Regression Tree and Gradient Boosted Trees. Each model’s performance was compared using the evaluation metrics such as accuracy and error.

Model Comparison:

1. Random Forest:

This model delivered the highest accuracy with an R2 value of 0.928. It also had the lowest error rates across the parameters, including mean absolute error (0.27) and root mean squared error (0.498). This consistency highlights its robustness and sustainability for the task

1. Simple Regression Tree:

The simple tree model had the lowest accuracy with an R2 value of 0.866. It also had the highest error rates such as a mean squared error of 0.458 and root mean squared error of 0.677. These results indicate its limited capacity in capturing complex patterns compared to other models. Inversely, low scores for simpler models like the Simple Regression Tree indicate the ETA prediction problem to be of a complex nature.

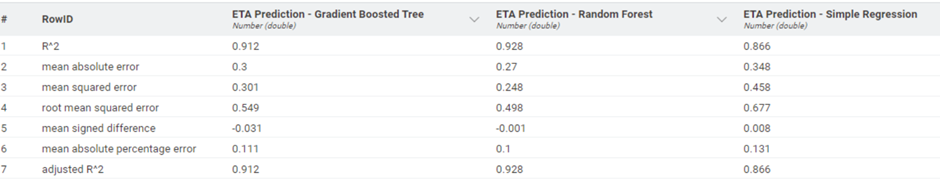
1. Gradient Boosted Trees:

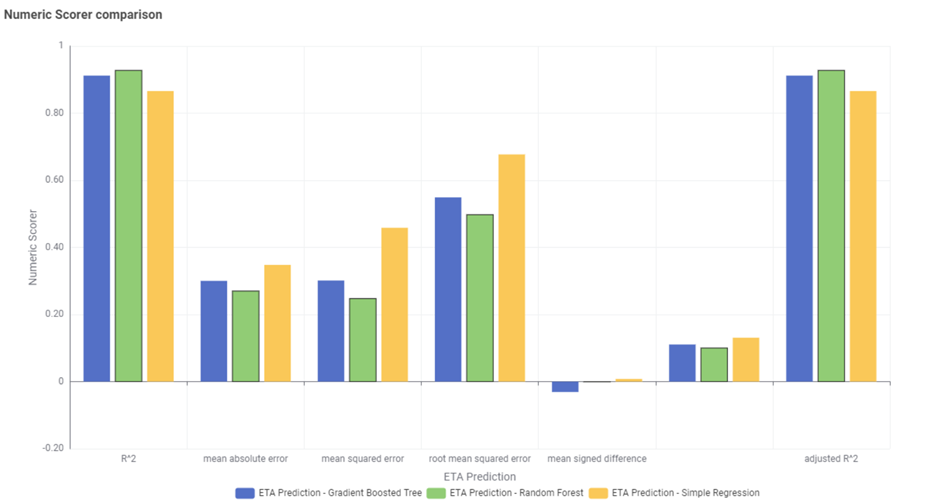
This model achieved an R2 of 0.912, slightly lower than Random Forest. While the performance was still competitive, its errors were slightly higher than the top-performing models.

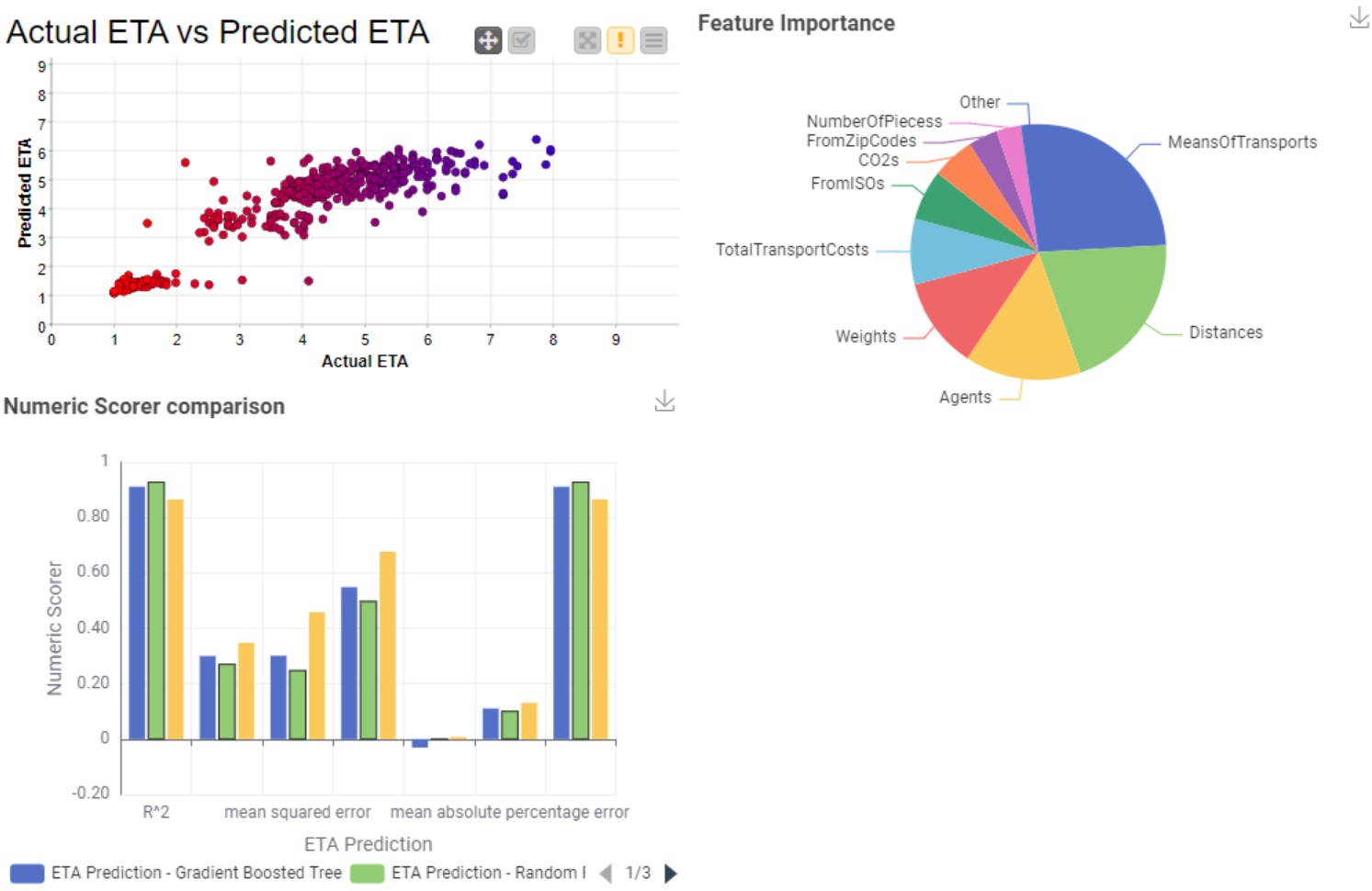
Key Takeaways:

* The Random Forest model is recommended for its superior accuracy and lower error rates.
* Gradient Boosted Trees provide strong alternatives with slightly lower performance but are still suitable for many applications depending on priorities.
* The Simple Regression Tree model was the weakest performer and may not be suitable for complex datasets like the TKL Logistics dataset.

Overall, this evaluation demonstrates the value of comparing multiple models to select the most effective solution for predictive analytics tasks in logistics data.







Based on our work on the TKL Dataset, we can understand the following:

* Within the data, we have several useful parameters which can be feature engineered to represent the same information with lesser parameters and can be quite useful in predicting the expected ETA of an order. There are a few parameters that are irrelevant to the problem and may be dropped during processing but otherwise the data itself is ripe and reliable for results.
* The data provided by TKL Logistics is in the generic JSON (JavaScript Object Notation) format and can be accessed through JSON Readers for KNIME or JSON related directories or support systems on most languages. From a GUI perspective, the data can be opened on most interfaces such as Notepad, WordPad, Google Chrome, or similar browsers.
* To structure the data for our purpose, we drop all values that do not affect the ETA, such as LogEntryID, ForwardingAgentID, EntryDate, UnloadingDate, ForwardingAgentNames and Agreement which can be avoided altogether to make the processing faster and more resource-efficient. Additionally, the LoadingDate and DeliveryDate can be replaced with a TotalDuration value that is the number of days between loading and delivery, thus reducing the data further. The TotalTransportCost can be converted to just a numeric value after dropping the currency representation at the end since all costs are in the same currency. After reduction, the names and other non-numeric values in data must be replaced to numeric representations since models can only learn from numeric values. Finally, we normalise the data to values between 1 and 10 so that larger numbers such as Transportation Cost or Zip Codes do not influence the model in an unintended manner due to its raw size.
* The initial data provided by TKL Logistics contains several entries that have questionable data, such as dates that are beyond 2024, or distances that are zero, or missing values in many parameters such as ETA and Cost. These outliers and missing values must be cleaned based on what the data can range between, what it represents, and what all parameters can influence the data. Basically, the data is to be cleaned using methods derived from relevant domain knowledge. There will be some unclean entries even after extensive cleaning, so we will drop these entries and resort to using only valid values to train the model
* After all the processing has been done, of the initial 82,741 historical entries in the dataset, we have **X** data entries that can be used to visualise the connections between data and train the model to predict the expected ETA.
* From the realtime data, we need only to use FromISO, FromZipCode, ToISO, ToZipCode, OriginalETA, NumberOfPieces, MeansOfTransport, Distance, Weight and Agent to calculate the expected ETA of the shipment.
* To deploy the models created in this project, the easiest and simplest solution would be to save the model and host the model as an API endpoint on a Python FastAPI/Flask script so that the client’s WebPage can invoke this model remotely and predict an expected timeline of delivery for their clients on the go. This would require the following specifications as minimum hardware requirements:
  + RAM: 2-4GB
  + Dual Core CPU with atleast 2.0GHz Clocking Speed
  + Storage: 5-10GB of Hard Disk Drive or Solid State Drive memory
  + Reliable network
* Additionally, the software requirements to run these models are as follows:
  + Python 3.8 or higher for FastAPI or Flask
  + Directories such as Scikit-Learn, Uvicorn, JSON and Flask or FastAPI directories.
  + Deployment Tools such as Docker
* To identify entries that are returns, a new column can be added based on the TotalTransportCost since the return entries are labelled with zero or negative values in the TotalTransportCost parameter.

- Suggest how selection can be made better with less CO2 emissions and faster ETA on the website