



ROUTE 25 - November

Ricardo Abreu

**14.11.2025**

8.11 November 2025

**Code available at:** [ricardo-abreu1/Route-25](https://github.com/ricardo-abreu1/Route-25)

8.11.1 Technical Activity – QoS Prediction Model Development

# **Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
| 11/11/2025 | 1 | Ricardo Abreu | Onboarding, 8.11.1 - Explorating QWS dataset |
| 12/11/2025 | 2 | Ricardo Abreu | 8.11.1 - Data Exploratory Analysis, Data Processed. Implemented LinReg, Random Forest and XGBoost |
| 13/11/2025 | 3 | Ricardo Abreu | 8.11.1 - Data Profiling, Exploratory Analysis, Data Processed (scripts improvements) |
| 14/11/2025 | 4 | Ricardo Abreu | 8-11-1 - Models Tuning and Evaluation |

**Table of Contents**

Contents

[Change History 2](#_Toc215226105)

[Week 1 4](#_Toc215226106)

[Load and Inspect Data sources – Dataset “QWS Dataset V1” 4](#_Toc215226107)

[Machine Learning model making: 10](#_Toc215226108)

[Confidence Interval 13](#_Toc215226109)

[Model Tuning - RandomizedSearchCV 14](#_Toc215226110)

[QQ plot: 19](#_Toc215226111)

[Feature Importance (Regression models): 20](#_Toc215226112)

[Week 2 22](#_Toc215226113)

[Classification Models 23](#_Toc215226114)

[Week 3 38](#_Toc215226115)

[Investigation links: 38](#_Toc215226116)

[Regression models 40](#_Toc215226117)

[Classification models 48](#_Toc215226118)

# Week 1

## Load and Inspect Data sources – Dataset “QWS Dataset V1”

A screenshot of a computer

AI-generated content may be incorrect.

The data source was in **.txt format** and contained comments and non-standard headers, which required initial cleaning. In this first step, a copy of the file was created, and the first 42 lines—containing only the project description—were removed. Next, a header row with column names was added to match the data structure. The file was then loaded into the local environment for data analysis.

A screenshot of a computer program

AI-generated content may be incorrect.

The next phase consisted of **exploratory data analysis** to gain a deep understanding of the data, metadata, and dataset structure.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

After that, the data was inspected for **null values, duplicates, and outliers**, following standard procedures for dataset profiling.

A screenshot of a computer screen

AI-generated content may be incorrect.

As a final note for this stage, the following points were observed:

1. Data types are correct;
2. The dataset contains no duplicates or null values;
3. The dataset includes some outliers; however, given its small size, these were retained.
4. The target variable chosen for prediction is **“WsRF: Web Service Relevancy Function (%)”**.

A screenshot of a computer program

AI-generated content may be incorrect.

In this phase, a **graphical analysis** was performed to better understand the data distribution and relationships between variables.  
Univariate, bivariate, and multivariate analyses were conducted, considering service classification into predefined categories.

Initially, the variables **“Response Time”** and **“Latency”** were explored due to their less symmetric distributions. Subsequently, the remaining numeric variables were analyzed according to their distribution.

A graph of a number of blue bars

AI-generated content may be incorrect.

A graph of blue and black bars

AI-generated content may be incorrect.

A graph of different colored bars

AI-generated content may be incorrect.A screenshot of a graph

AI-generated content may be incorrect.

The next step involved creating a **correlation matrix** during the bivariate analysis to identify patterns among variables.

A screenshot of a computer

AI-generated content may be incorrect.

A graph with numbers and symbols

AI-generated content may be incorrect.

Below is an example of a chart illustrating the decision to drop a variable from the model because it showed no meaningful pattern with the target variable.

A screen shot of a computer program

AI-generated content may be incorrect.

A graph of lines and colors

AI-generated content may be incorrect.A graph with blue dots

AI-generated content may be incorrect.

Finally, a brief multivariate analysis was performed to understand how certain characteristics relate to service level.

Key findings from this analysis:

* Univariate Analysis:
  + Availability is concentrated near 100%, suggesting most services are highly available.
  + Successability, Reliability, and WsRF are fairly normal, centered around 50–80.
* Bivariate Analysis:
  + Platinum services (Class 1) generally have lower Response Time and Latency compared to Bronze.
  + Throughput tends to decrease as class level increases (Platinum → Bronze).
  + Reliability and Successability are higher for Platinum and Gold.
  + Response Time and Latency show strong positive correlation (services with high response time also have high latency).
  + Best Practices vs WsRF shows weak correlation, suggesting WsRF may not strongly depend on Best Practices.
  + Response Time and Latency have the highest correlation among numeric features.
  + Availability and Documentation have low correlation with other metrics, indicating limited predictive power.
* Multivariate Analysis:
  + Platinum services cluster with low Response Time and Latency, high Reliability and Successability.
  + Bronze services show the opposite pattern: high delays and lower reliability.
  + Columns that can be dropped - (Latency or Response Time, only one) and Best Practices.

## Machine Learning model making:

In this phase, **Machine Learning (ML) models** were built to train on the data and predict future datasets.  
The workflow followed this sequence:

1. Split the data into training and test sets.
2. Define preprocessing steps. For the Linear Regression model, the data suffers an additional step of “scaling“ so that the features had a similar distribution.
3. Train the model using the training set.
4. Evaluate the model on the test set.
5. Tune the model (an iterative process to identify the best parameters for improved performance).

Data Splitting:

The figure below illustrates how data was split and columns with low correlation were removed (to reduce noise in the model).

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

The data was split into 3 distinct subsets. The first subset is the training data that will be used to train the ML model, the second is test data to test the ML model according the certain metrics and finally the third subset is the validation data that it wont be used in this task but in a different task in the future for back testing and data drifts.

Next, a **baseline model** was defined for comparison with subsequent models. There is 2 ways to do this. The first picture shows the manual way to compute metrics for the baseline model and the second figure shows how to do it according to scikit-learn.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

The models selected were **Linear Regression, Random Forest, and XGBoost**.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

The following table shows the results of 3 metrics for all 3 models:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | RMSE | MAE | R2 |
| Linear Regression | 3.5986977509976765 | 2.6328531950906426 | 0.927806724857026 |
| Random Forest | 3.8037618366018973 | 2.5113698630136994 | 0.9193447515586426 |
| XGBoost | 3.8115673065185547 | 2.6854708194732666 | 0.9190133810043335 |

A graph with different colored bars

AI-generated content may be incorrect.

## Confidence Interval

A confidence interval gives a range of values that likely contains the true metric for the model, based on testing data. For it was applied bootstrapping. Bootstrapping simulates many possible test sets by resampling testing data.

A screen shot of a computer program

AI-generated content may be incorrect.

A graph of a graph with numbers and a line

AI-generated content may be incorrect.A graph with a line graph and numbers

AI-generated content may be incorrect.

A graph of a graph with numbers and a line

AI-generated content may be incorrect.

**1. Linear Regression** (benefits greatly from more data, overfitting is evident at small sample sizes)

    - Train RMSE (blue): Starts near zero and stays extremely low across all training sizes.

    - The model fits training data almost perfectly, which is expected for linear regression on small datasets.

    - CV RMSE (orange): Starts very high (~193) with small training size, then drops sharply to ~3.3 at full size.

    - Gap between train and CV RMSE at small sizes > severe overfitting initially.

    - Trend: As training size increases, CV RMSE decreases and stabilizes but never meets train RMSE.

**2. Random Forest** (highly flexible and overfits small datasets, dding data improves CV RMSE but gap persists)

    - Train RMSE (blue): Very low (~1.5) and flat across all sizes > model memorizes training data.

    - CV RMSE (orange): Starts at ~20.6, decreases gradually to ~6.0 at full size.

    - Shaded region: Wide at small sizes, narrows as size grows > variance reduces with more data.

**3. XGBoost** (Overfits small datasets, Still needs regularization or more data to close the gap)

    - Train RMSE (blue): Near zero across all sizes > extreme memorization.

    - CV RMSE (orange): Starts at ~19.2, drops to ~6.0, overfitting similar to Random Forest but better CV RMSE at mid-range sizes.

    - Variance: High at small sizes, narrows with more data.

## Model Tuning - RandomizedSearchCV

Nesta secção é descrito como os modelos foram melhorados. Cada modelo tem um algoritmo com hyper-parâmetros (parâmetros do algoritmo) que podem ser modificados no sentido de melhorar a performance por parte do modelo de ML. Segue o exemplo abaixo para descrever parâmetros que podem ser mudados na Regressão Linear:

Eg. Hyperparameters for Linear Regression:

* - fit\_intercept: Whether to calculate the intercept (True/False).
* - positive: Force coefficients to be positive (True/False).
* - normalize: Deprecated in newer versions (use StandardScaler instead).
* - n\_jobs: For parallel computation (usually not critical here)

A black screen with white text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Afterwards, the train data was fed into the models to predict values.

A screenshot of a computer program

AI-generated content may be incorrect.

After calling the function to predict the values, all of the aspects related to evaluating the models were made.

A black screen with white text

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

Then it was produced a sample of the dataframe with actual values and the ones predicted by each model.

A screenshot of a computer

AI-generated content may be incorrect.

Afterwards, all metrics of all models were graphically put together for visibility,

A computer screen shot of a program code

AI-generated content may be incorrect.

A graph of different colored bars

AI-generated content may be incorrect.

* **RMSE (Root Mean Squared Error):** Measures prediction error magnitude. Lower is better.
* **MAE (Mean Absolute Error):** Measures average absolute deviation. Useful for interpretability.
* **R² (Coefficient of Determination):** Indicates how much variance in the target is explained by the model. Closer to 1 means better fit.

Then, the results from the tuning of the models, the results from the models that were not tuned and the baseline model were compared, as showed in the previous picture. It’s results are in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | RMSE | MAE | R2 |
| Baseline | 13.408000765060446 | 10.687379372028431 | -0.002153048229032839 |
| Linear Regression | 3.5986977509976765 | 2.6328531950906426 | 0.927806724857026 |
| Random Forest | 3.8037618366018973 | 2.5113698630136994 | 0.9193447515586426 |
| XGBoost | 3.8115673065185547 | 2.6854708194732666 | 0.9190133810043335 |
| Linear Regression (tuned) | 3.5986977509976765 | 2.6328531950906426 | 0.927806724857026 |
| Random Forest (tuned) | 3.8263169789963936 | 2.513851973011654 | 0.918385393789493 |
| XGBoost (tuned) | 2.7543282508850098 | 1.8014215230941772 | 0.9577100276947021 |

This next code block produced the following:

A screen shot of a computer program

AI-generated content may be incorrect.

A graph of a distribution

AI-generated content may be incorrect.A graph with blue dots and red line

AI-generated content may be incorrect.

**For the Residual Plot (Predicted vs Residuals):**

* Centering around zero: Most residuals cluster near the red line (0), which is good — predictions are generally unbiased.
* Spread increases slightly for higher predicted values: At predicted values above ~75, residuals become more positive and more variable.
* This suggests mild heteroscedasticity (variance of errors grows with predicted value).
* Outliers: There are a few large positive residuals (up to ~15), meaning the model underpredicted those cases significantly.

**Residual Distribution (Histogram):**

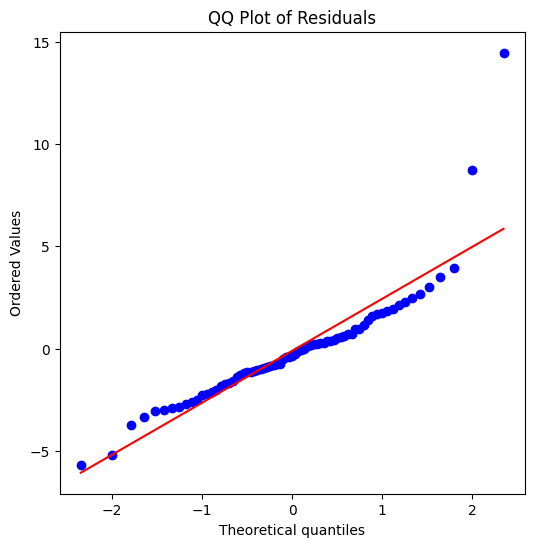
* Shape: Mostly bell-shaped and centered near zero, which is ideal for regression assumptions.
* Slight right skew: A tail toward positive residuals indicates some cases where the model underestimates the target.
* No severe multimodality: No clear clusters, so the model likely captures the main pattern well.

## QQ plot:

The following picture shows the code block for the QQ Plot.

A screen shot of a computer program

AI-generated content may be incorrect.



Central Alignment:

* Most points lie close to the red line in the center, meaning residuals are approximately normal for the bulk of the data.

Heavy Right Tail (Positive Outliers):

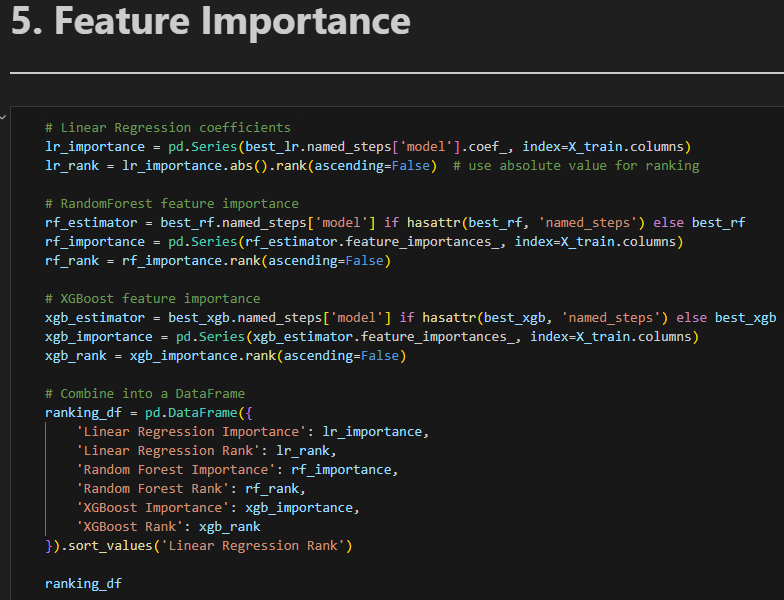
* The top-right points deviate strongly from the line, especially two extreme points around 9 and 15.
* This indicates large positive residuals. Model underpredicted these observations significantly (normal because since the amount of data is small the outliers were kept)

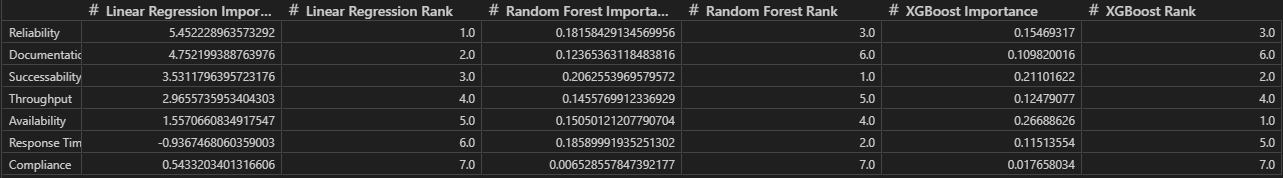
Slight Left Tail Deviation:

* Bottom-left points also deviate, but less severely than the right tail.
* Suggests some negative outliers (overpredictions), but not as extreme.

## Feature Importance (Regression models):

In this section it’s described the steps to find the importance of the features for each model. So, the next 2 figures explain how this was done, and then presented a table with the features and their respective importance for the models ordered by the Linear Regression Ranking.





# Week 2

To finish the script of the regression models, a final graph was constructed showing the features importances according to a Shap graph for the model with the best results which was XGBoost.

A screen shot of a computer program

AI-generated content may be incorrect.

A graph of different colored dots

AI-generated content may be incorrect.

Finally, the models were exported, with all the metrics for evaluation for json and csv format, and also a md file was created depicting at a high level the model and its metrics as showed in the following figures.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

## Classification Models

To construct a script for a categorical model, the process is almost the same as the regression, so the script follows almost the same structure. One of the key differences in this script is that the label (target value) which currently is a class between 1 through 4 already in the datatype integer). In this example it is not necessary but for consistency it suffered a process of encodement just in case any data was different than integer. Then this column fits these values and assigns values from 0 through 3, this is important because some ML Models work with 0 as a starting point.



Following a similar matter, the data was split into 3 subsets (training data, test data and validation data). This procedure is similar to the regression models depicted earlier.

A screen shot of a computer program

AI-generated content may be incorrect.

Then a baseline model was established to predict the most frequent class.

A screenshot of a computer program

AI-generated content may be incorrect.

Just to have an idea of the predicted values of the classification models chosen, the following pipeline was constructed, and the values are in the following figures.

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

A black and white screen with white text

AI-generated content may be incorrect.

Following the same procedure for the regression models, these were evaluated according to 5 of the following metrics.

A screenshot of a computer

AI-generated content may be incorrect.

A graph of different colored bars

AI-generated content may be incorrect.A computer screen shot of a program code

AI-generated content may be incorrect.

These next sections are for the tuning of hyperparameters for all models.

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

A screen shot of a computer program

AI-generated content may be incorrect.

After tunning the models, by settings hyperparameters that influence the model, a cross-validation was made to identify the best model overall based on F1 metric.

A screenshot of a computer

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

For results, it was shown the best model was the tuned Logistic Regression the following details:

* Accuracy (0.8405 ± 0.0543) -> The model correctly predicts about 84% of cases on average. The ±0.05 shows moderate variability across folds, which is acceptable for this dataset size.
* Precision\_macro (0.8438) -> When the model predicts a class, it is correct 84% of the time on average across all classes, indicating reliable predictions.
* Recall\_macro (0.8549) -> The model captures about 85% of actual cases, showing strong sensitivity and ability to identify true positives.
* F1\_macro (0.8469 ± 0.0506) -> A balanced measure of precision and recall. At ~85%, the model demonstrates excellent overall performance.
* ROC\_AUC\_OVR (0.9669) -> Outstanding ability to separate classes (close to 1.0). This suggests the model is highly effective at distinguishing between different service quality levels.

In these 2 next figures, it’s showed an example of predicted values by every model and the actual values and the metrics for all models respectively.

It is important to note that, in each day the code was reviewed and optimized some sections according not only to the current task but also for future ones. Nontheless in this report it’s showed everything for transparency and to show the progress of the author.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

As the regression model, it was made a graph with a learning curve for the tuned logistic regression classification model.

A screen shot of a computer program

AI-generated content may be incorrect.

A graph showing a long line

AI-generated content may be incorrect.

The learning curve illustrates how the F1 score (macro) evolves as the training set size increases:

* Initial Phase (≈94 samples): The training score starts very high (≈0.95), while the validation score is low (≈0.34). This large gap indicates strong overfitting when the model is trained on a small dataset — it memorizes the training data but fails to generalize.
* Intermediate Phase (≈160–225 samples): As more data is added, the validation score improves significantly (rising to ≈0.82), while the training score decreases slightly (≈0.81). This convergence suggests the model is learning more generalizable patterns and reducing overfitting.
* Final Phase (≈291 samples): Both curves stabilize close together (training ≈0.90, validation ≈0.85). The small remaining gap indicates mild overfitting, which is expected given the limited dataset size. The plateau in the validation curve suggests that adding more data may yield only marginal improvements unless combined with feature engineering or regularization adjustments.
* Confidence Intervals: The shaded regions show variability across folds. Wider intervals at smaller training sizes reflect instability due to limited data, while narrower intervals at larger sizes indicate more consistent performance.

After this, confusing matrixes were constructed to further evaluate the values predicted and the actual values of the test dataset.

A screen shot of a computer program

AI-generated content may be incorrect.

A close-up of a graph

AI-generated content may be incorrect.After this, confusing matrixes were constructed to further evaluate the values predicted and the actual values of the test dataset.

A close-up of a graph

AI-generated content may be incorrect.

A close-up of a graph

AI-generated content may be incorrect.

Each confusion matrix shows actual values vs predicted classes. The left matrix displays raw counts, while the right matrix shows percentages normalized per actual class. This helps identify which classes are predicted accurately and where misclassifications occur relative to class size.

Following the same analysis of the previous type of models (regression), a feature importance analysis was made for all tuned classification models.

A screen shot of a computer program

AI-generated content may be incorrect.

A graph with blue bars

AI-generated content may be incorrect.

A graph of blue bars

AI-generated content may be incorrect.

A graph of blue bars

AI-generated content may be incorrect.

**1) Random Forest Tuned Importance:**

    - Top Feature: Reliability is the most influential feature, indicating that service reliability strongly impacts the predicted quality level.

    - Other High-Impact Features: Throughput and Successability follow closely, suggesting performance and success rate are critical.

    - Moderate Importance: Documentation and Response Time still play a role but less than reliability.

    - Least Important: Compliance and Best Practices have minimal impact, meaning they contribute little to prediction accuracy.

**2) XGBoost Tuned Importance:**

    - Top Feature: Availability dominates, showing that uptime is the strongest predictor for XGBoost.

    - Balanced Importance: Reliability, Successability, and Throughput are all significant, reinforcing the importance of performance metrics.

    - Moderate Role: Documentation and Response Time are secondary factors.

    - Least Important: Compliance and Best Practices remain low, similar to Random Forest.

**3) Logistic Regression Importance:**

    - Top Feature: Latency has the highest coefficient magnitude, meaning delays strongly influence the predicted class.

    - Second Strongest: Response Time also matters significantly.

    - Moderate Impact: Reliability and Documentation contribute meaningfully.

    - Low Impact: Best Practices and Compliance have very small coefficients, confirming their limited predictive power.

**Overall:**

- Across all models, features tied to speed (Latency, Response Time) and stability (Reliability, Availability) consistently rank high.

- Documentation is Secondary: While not negligible, documentation is less critical than technical performance.

- Compliance & Best Practices Are Weak Predictors consistently rank lowest, suggesting they do not strongly differentiate service quality levels.

After this a SHAP graph was made for all tuned models as per identified in the following pictures.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen with text on it

AI-generated content may be incorrect.

A graph of a graph with red and blue dots

AI-generated content may be incorrect.

A graph of different colored dots

AI-generated content may be incorrect.

A graph of blue and pink dots

AI-generated content may be incorrect.

**Top Influential Features Across Models:**

* Response Time and Successability consistently show the strongest impact on predictions.
* Throughput and Availability also contribute but to a lesser extent.
* Interaction Effects: SHAP interaction plots indicate weak feature interactions overall, meaning predictions rely on combined effects rather than a single dominant feature.

**Model-Specific Observations:**

* Random Forest and XGBoost emphasize performance and reliability metrics.
* Logistic Regression highlights latency-related features (e.g., Response Time).

Interpretability Insight: No single feature overwhelmingly drives predictions; instead, multiple features collectively influence the outcome, reinforcing the need for a balanced feature set.

To end the classification models for the first task, the best model was exported with all metrics into a json and csv file, and also a md file was constructed following the same procedure as the regression models.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

# Week 3

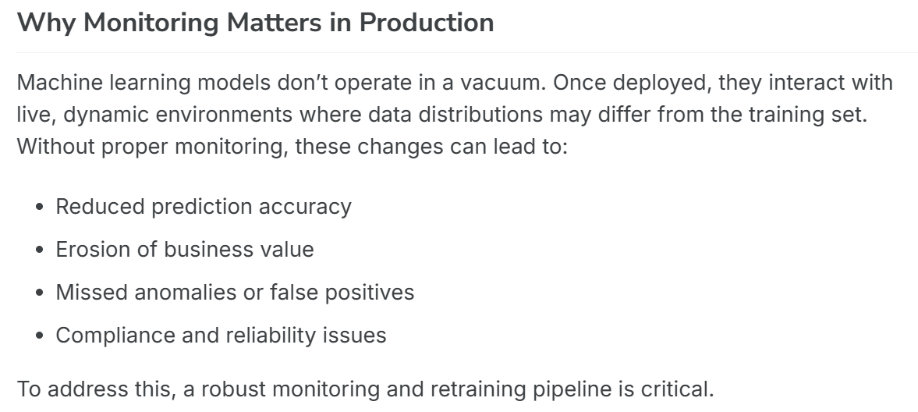
Investigation links:

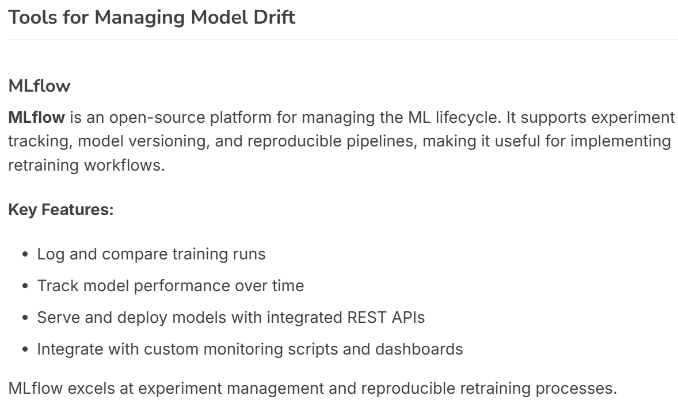
1. [How to Build Model Monitoring Systems in Production: A Step-by-Step Guide | Codez Up](https://codezup.com/building-model-monitoring-systems-in-production-guide/#:~:text=In%20this%20guide%2C%20you%E2%80%99ll%20learn%20how%20to%20build,like%20Flask%2C%20Prometheus%2C%20and%20Grafana.%202.%20Technical%20Background)

A screenshot of a computer

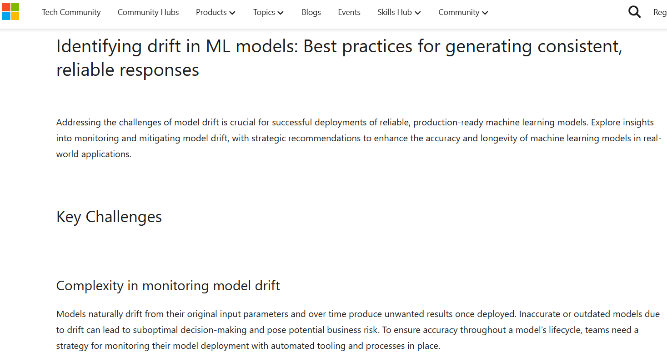
AI-generated content may be incorrect.

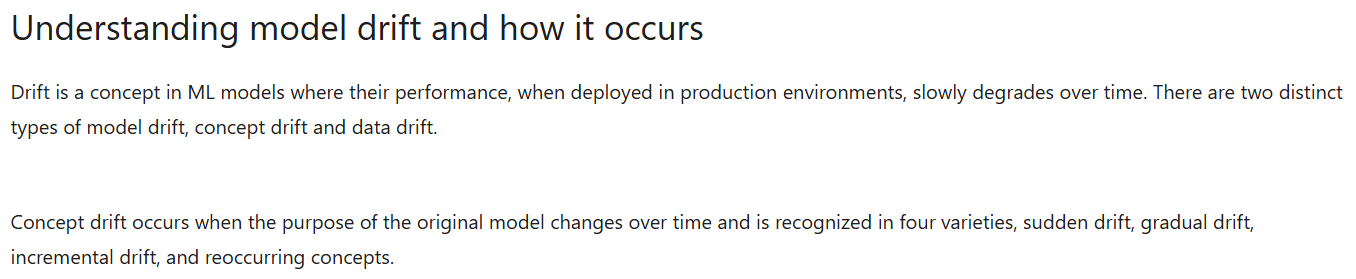
2. [Techniques for Monitoring and Managing Model Drift in Production - Data Science](https://diogoribeiro7.github.io/machine%20learning/model%20monitoring/techniques_moniitoring_managing_model_drift_production/)

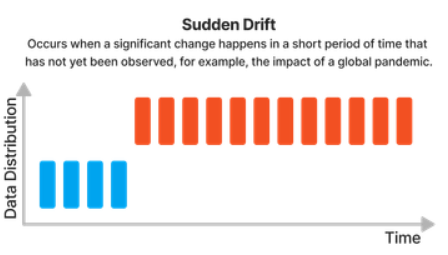
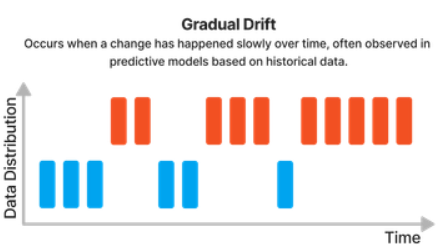


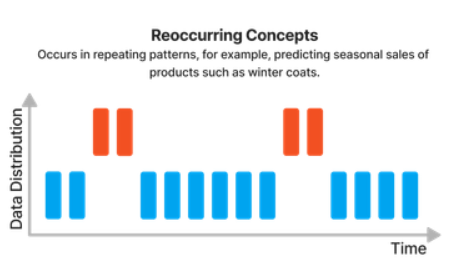


3. [Identifying drift in ML models: Best practices for generating consistent, reliable responses](https://techcommunity.microsoft.com/blog/fasttrackforazureblog/identifying-drift-in-ml-models-best-practices-for-generating-consistent-reliable/4040531)

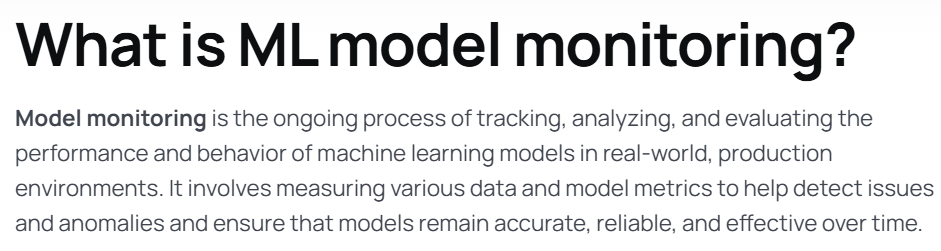


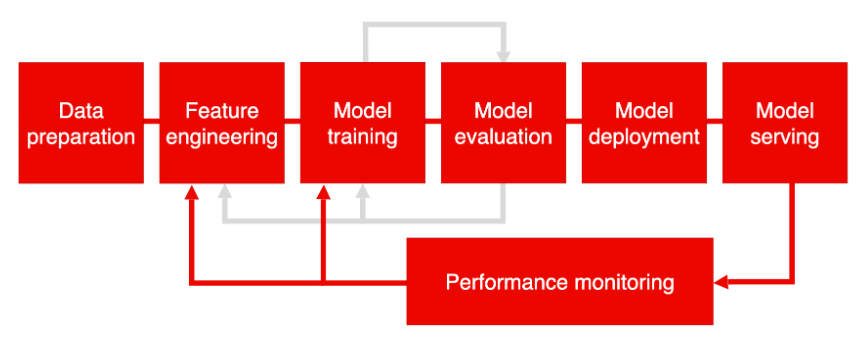


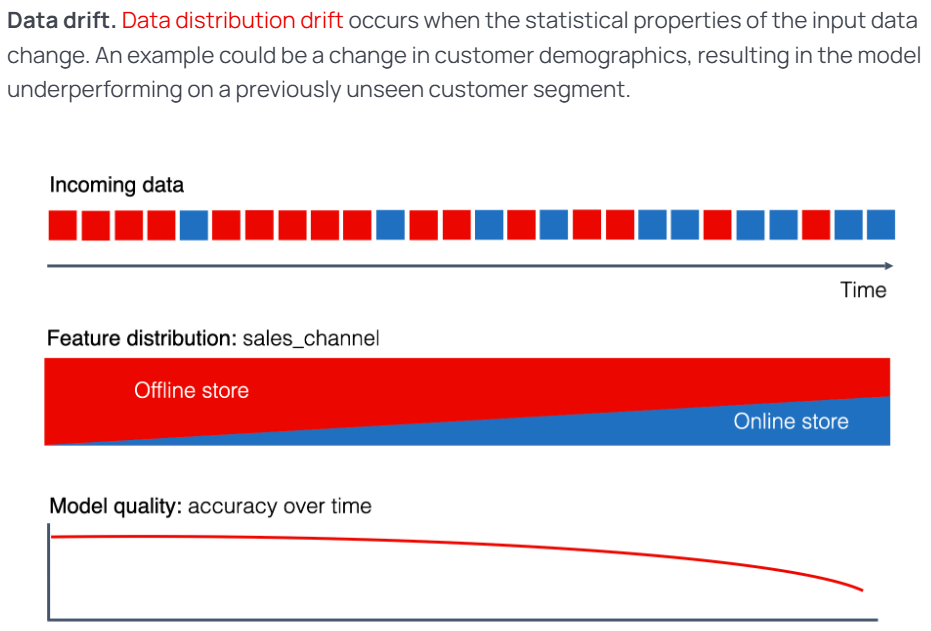




4. [Model monitoring for ML in production: a comprehensive guide](https://www.evidentlyai.com/ml-in-production/model-monitoring)







## Regression models

For the following task, it is necessary to do the following procedure:

* + Define official metric to evaluate the model
  + Implement k-fold or time-based back tests (in this case we will do k-fold back tests since the data doesn’t have time base data)
  + Generate plots for data analysis
  + Data drift checks
  + Report summary

The regression models were evaluated first. In this case the defined metrics were R2 for the primary metric and then for secondary metrics RMSE and MAE in order to give a more overall overview of the results. For this, the tuned models from the previous task were loaded into the script as the picture shows.

A screen shot of a computer program

AI-generated content may be incorrect.

Then the back-test was started by dividing the validation data (data split from the original dataset that wasn’t trained by the model or evaluated, in other words it’s data completely new to the model). The idea is to divide the data into k (number) equal parts, train the model on the current k split of the data and test the model on the remaining folds. This process is iterative to finally receive an average performance across all k folds of data. The next figure shows the configuration steps prepared.

A computer screen with text on it

AI-generated content may be incorrect.

Then finally, in the next figure, shows the split strategy for the k folds, and the code required in order to do the back test for the tuned regression models.

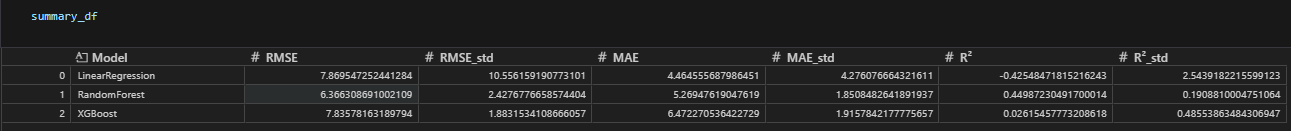
A screen shot of a computer program

AI-generated content may be incorrect.

The next 2 figures show the results for each fold and the average for each metric decided previously.

A screenshot of a computer

AI-generated content may be incorrect.



In the first picture it shows results for the metrics of each fold resulting in the following aspects:

1. **Linear Regression**

* RMSE: Ranges from 2.75 (Fold 1) to 26.75 (Fold 2), indicating extreme variability.
* R2: From 0.92 (good fit) to -4.95 (very poor fit).
* Interpretation: The model is highly unstable across folds. Negative R2 values mean the model performs worse than a simple mean predictor in some folds. This suggests Linear Regression cannot capture the complexity of the data and is sensitive to data splits.

1. **Random Forest**

* RMSE: Between 2.51 (Fold 3) and 8.75 (Fold 5), much more consistent than Linear Regression.
* R2: Positive across all folds (0.21–0.64), showing moderate predictive power.
* Interpretation: Random Forest is robust and stable, handling variability better than Linear Regression. It adapts well to non-linear relationships.

1. **XGBoost**

* RMSE: From 3.47 (Fold 3) to 9.27 (Fold 5), slightly higher than Random Forest in some folds.
* R2: Varies widely (0.48 in Fold 1 to -0.64 in Fold 4), including negative values.
* Interpretation: XGBoost shows inconsistent performance. While it performs well in some folds, negative R2 in others indicates overfitting or sensitivity to certain data partitions.

After the back-testing, some plots for data analysis were made to evaluate the model on unseen data (validation data). The next figures show the code produced to construct these graphs. First some configuration aspects, then the actual code for the plots.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

The previous figures produced the following graphs for each of the tuned ML models.

A graph with blue dots

AI-generated content may be incorrect.

A graph with dots and lines

AI-generated content may be incorrect.

A graph with dots and lines

AI-generated content may be incorrect.

This results in the following analysis.

**1. Linear Regression Tuned**

* Predicted vs Actual Plot:
* Points are close to the diagonal red line, indicating good alignment between predictions and actual values.
* R2 = 0.916 - The model explains 91.6% of variance.
* Slight spread at higher values suggests minor under/over-prediction at extremes.
* Residual Plot
* Residuals are scattered around zero without a clear pattern, no strong heteroscedasticity.
* However, some residuals reach ±6, indicating occasional large errors.
* Error Distribution Plot
* Histogram is roughly centered around zero but slightly skewed to the positive side.
* Most residuals fall between -4 and +4, with a few outliers.
* Shape is close to normal.

**2. Random Forest Tuned:**

* Predicted vs Actual Plot
* Points are tighter around the diagonal compared to Linear Regression.
* R2 = 0.938, RMSE and MAE are lower - better performance than Linear Regression.
* Slight clustering near higher actual values, but overall strong fit.
* Residual Plot
* Residuals are more evenly distributed than Linear Regression, but some negative residuals at lower predicted values.
* No obvious trend, which is good.
* Error Distribution Plot
* Centered near zero, slightly narrower spread than Linear Regression.
* Fewer extreme residuals - Random Forest handles variability better.

**3. XGBoost Tuned:**

* Predicted vs Actual Plot
* Points almost perfectly align with the diagonal.
* R2 = 0.961, RMSE = 2.01, MAE = 1.58 - best performance among all models.
* Very tight clustering, minimal deviation.
* Residual Plot
* Residuals are small and evenly scattered around zero.
* No visible pattern.
* Error Distribution Plot:
* Very narrow spread, most residuals between -3 and +3.
* Almost symmetric around zero – strong indication of unbiased predictions.

After this, an analysis of the data drift using the trained tuned models and the validation data to understand how the model acts on unseen data. The first figure shows the code to produce all the metrics necessary to conduct an analysis of this kind. The second figure shows the results in a table for organizational purposes. Just to clarify some information, the training data had 254 rows, validation data had 33 rows and for PSI thresholds – less than 0.1 (No drift), from 0.1 to 0.25 (Moderate drift) and more than 0.25 (Significant drift).

A black table with white text

AI-generated content may be incorrect.

The top three features from, according to the PSI values are shown in the following figures.

A computer screen shot of a program code

AI-generated content may be incorrect.

A graph of a curve

AI-generated content may be incorrect.

A graph of a diagram

AI-generated content may be incorrect.

A graph of a graph

AI-generated content may be incorrect.

1. Reliability

* Validation distribution is shifted to the right compared to training.
* Indicates higher reliability values in validation data.
* PSI confirms this as the most drifted feature.

1. Throughput

* Validation curve shows heavier tail on the right.
* Suggests more high-throughput observations in validation compared to training.
* This could affect model predictions if throughput strongly influences the target.

1. Successability

* Validation distribution slightly shifted right.
* Indicates improved successability in validation data.
* Drift is significant but less extreme than Reliability

All these results were saved for later use as part of the Route 25 project.

## Classification models

A similar procedure was conducted for these types of ML models. The differences will be depicted in this document. As the first step, definition of metrics, the following are the metrics considered for evaluating - **Primary:** F1-score, **Secondary:** Accuracy, Precision, Recall, AUC – which will be evaluated for all 3 tuned models.

For the back-test, the next figures it is shown the code used for the test, the metrics for each fold (subset of the validation data). There is also another table that aggregates these values by the average for each mode after the figures.

A computer screen shot of many colorful text

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

A table with numbers and letters

AI-generated content may be incorrect.

After the back-testing some graphs were constructed to evaluate the tuned classification ML models, although in this case, it was built confusion matrices, ROC curves and PR (precision-recall) curves.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

1. **Confusion Matrices:**

* Logistic Regression Tuned
* Most predictions are correct for classes 2 and 3.
* Minor misclassifications occur between classes 0 and 1.
* Random Forest Tuned
* Performs well on class 2 but shows more confusion between classes 0 and 1 compared to Logistic Regression.
* Class 3 predictions are slightly less accurate than Logistic Regression.
* XGBoost Tuned
* Similar to Random Forest, strong performance on class 2, but some misclassification between classes 0 and 1.
* Overall balanced but slightly less precise than Logistic Regression for class 3.

Logistic Regression shows the most consistent diagonal dominance (better separation), while tree-based models have more overlap between lower classes.

1. **ROC Curves:**

All models show AUC values close to 1.0 for most classes, indicating strong discriminative ability.

* Logistic Regression: Curves are steep and close to the top-left corner for classes 2 and 3 - excellent performance.
* Random Forest & XGBoost: Slightly less steep for class 1, meaning weaker separation for that class compared to Logistic Regression.

All models perform well overall, but Logistic Regression has the most consistent ROC curves across classes.

1. **Precision-Recall Curves**

* Logistic Regression: High precision across most recall levels for classes 2 and 3.
* Random Forest & XGBoost: More variability and lower precision for classes 0 and 1, especially at higher recall.

PR curves confirm that minority classes (likely class 0 or 1) are harder to predict accurately for tree-based models. Logistic Regression maintains better precision-recall trade-offs for dominant classes, while Random Forest and XGBoost struggle more with minority classes.

Following the same procedure as the regression models, a data drift analysis was conducted as shown in the following figures.

A screen shot of a computer program

AI-generated content may be incorrect.

A black and white table with numbers and letters

AI-generated content may be incorrect.

From the results we can infer the following:

Availability (PSI = 1.56), Successability (1.45), Throughput (0.94), Documentation (0.87), Best Practices (0.61), and Response Time (0.38) show major distribution shifts between training and validation. Latency (0.25) and Reliability (0.25) are borderline significant drift.

Stable Feature: Compliance (PSI = 0.09) shows no drift, its distribution remained consistent.

KS Test: KS p-values > 0.05 for all features -> shape differences are not statistically significant, but PSI highlights magnitude shifts.

Validation data differs substantially from training for key QoS features, which can affect model generalization. Monitoring or retraining may be needed.

A computer screen shot of a program code

AI-generated content may be incorrect.

A graph of a number of individuals

AI-generated content may be incorrect.

A graph of a graph

AI-generated content may be incorrect.

A graph of a curve

AI-generated content may be incorrect.

For the three top features, according to the PSI value, here are the following analysis:

* Availability:
  + Validation curve shifted right (higher values) compared to training.
  + Indicates services in validation set have better availability than training.
  + Matches PSI = 1.56 (largest drift).
* Successability:
  + Slight right shift for validation curve.
  + Suggests improved successability in validation data.
  + PSI confirms significant drift (1.45).
* Throughput:
  + Validation curve shows heavier tail on the right.
  + Indicates more high-throughput observations in validation.
  + PSI = 0.94 -> strong drift.

About Capgemini Engineering

World leader in engineering and R&D services, Capgemini Engineering combines its broad industry knowledge and cutting-edge technologies in digital and software to support the convergence of the physical and digital worlds. Coupled with the capabilities of the rest of the Group, it helps clients to accelerate their journey towards Intelligent Industry. Capgemini Engineering has 65,000 engineer and scientist team members in over 30 countries across sectors including Aeronautics, Space, Defense, Naval, Automotive, Rail, Infrastructure & Transportation, Energy, Utilities & Chemicals, Life Sciences, Communications, Semiconductor & Electronics, Industrial & Consumer, Software & Internet.

Capgemini Engineering is an integral part of the Capgemini Group, a global business and technology transformation partner, helping organizations to accelerate their dual transition to a digital and sustainable world, while creating tangible impact for enterprises and society. It is a responsible and diverse group of 340,000 team members in more than 50 countries. With its strong over 55-year heritage, Capgemini is trusted by its clients to unlock the value of technology to address the entire breadth of their business needs. It delivers end-to-end services and solutions leveraging strengths from strategy and design to engineering, all fueled by its market leading capabilities in AI, cloud and data, combined with its deep industry expertise and partner ecosystem. The Group reported 2023 global revenues of €22.5 billion.

Get the future you want | [www.capgemini.com](https://www.capgemini.com)

[](https://de-de.facebook.com/Capgemini) [](http://www.linkedin.com/company/capgemini) [](http://www.slideshare.net/capgemini) [](http://www.youtube.com/capgeminimedia)

This presentation contains information that may be privileged or   
confidential and is the property of the Capgemini Group.

Copyright © 2024 Capgemini. All rights reserved.