EP2420/EP272V Project 1
Estimating Service Metrics from Device Measurements

Rolf Stadler Xiaoxuan Wang

October 13, 2025

Project Objective

In this project, you train regression models that map IT and network infrastructure measurements X to predictions of service-level metrics Y. The measurements are obtained from a Video-on-Demand (VoD) service and a Key-Value store (KV) service, and the service-level metrics are Video Frame Rate and Response Time experienced by clients.

Using machine-learning techniques, the problem is to find a function (i.e., train a model) $M : X \to \hat{Y}$, such that $\hat{Y}$ closely approximates Y for a given X. Formally, we predict the expectation of $P(Y|X)$. If Y is numeric, the problem is referred to as a regression problem; if Y is a class or category, the problem is called a classification problem. This project considers both of these problems.

The machine-learning methods you will use are linear regression, random forest regression and classification, and neural network regression.

To estimate M, measurement pairs (or observations) of the form (x,y) are needed. A list of such measurement pairs ordered by time stamps is called a *trace*. The traces in this project are based on observations collected once per second during several hours from a running system.

You can find a description of the infrastructure and the measurements that produced the traces you use in this project in [1].

Project tasks

The project is composed of four mandatory tasks and two optional tasks.

You will be given traces collected from a KTH testbed [1]. Each trace includes two files, namely, the design matrix X and the target Y. They relate to two services, a Key-Values store service (KV) and a Video-on-Demand service (VoD). For the KV trace, use the target "ReadsAvg", and for the VoD trace, use the target "DispFrames". At deadline of each week, you submit two files, a report in pdf and a Jupyter notebook file with the code. Name your files Your_name.Task(s).pdf or Your_name.Task(s).ipynb, respectively. You put the two files in the same folder and upload the zipped folder on Canvas. For all numerical results in your reports, give three significant digits, for instance, 52.3 or 5.23e+01. For all tasks, provide a short description where you try to explain your understanding of the results from an engineering perspective, in addition to the data analysis perspective.

- Week 1: Task I
- Week 2: Task II
- Week 3: Task III
- Week 4: Task IV (Task V and/or Task VI) and final report
- Task VI can be submitted at any week.

Task I - Data Exploration and Pre-processing

1. You are given a trace with X_0, Y. Describe the design matrix X_0 in terms of the number of sample rows, and the number of feature columns. For the target values Y, compute the mean, standard deviation, maximum, minimum, 25th percentile, 50th percentile, and 95th percentile. Further, provide the density and histogram plots on the same figure for the target values.

2. Pre-process the data X_0 using the three methods below. Perform these methods on the feature columns as well as on the sample rows. This step produces six design matrices $X_1, X_2, ..., X_6$ in form of numpy arrays.

   (a) l^2 Normalization: linearly scale the values of each feature column so that its l^2-norm becomes 1 (X_1). Alternatively, linearly scale the values of each sample row so that its l^2-norm becomes 1 (X_2).

   (b) Restriction to interval: linearly map the values of each feature column (X_3) (or sample row (X_4)) so that all lie within the interval [0,1].

   (c) Standardization: map the values of each feature column (X_5) (or sample row (X_6)) linearly so that they have 0 mean and a variance of 1.

3. Reduce X_0 as follows. Use a tree-based method to select the top 18 features. This produces the design matrix X' and the corresponding target vector of Y. Now using X' and Y, we create the new matrix $Z = (X', Y)$ through horizontal concatenation. Create a correlation matrix of Z. This is a square matrix whose cells (i,j) show the correlation between feature column i and feature column j. Plot a heatmap of the correlation matrix. Describe and comment on your observations of this heatmap plot. **You will use the reduced matrix X' in Task II.**

4. From the above heatmap, choose two features, one with the highest correlation (close to 1) with the target and one with the lowest correlation (close to 0). We call the feature with the highest correlation feature F_h and the feature with the lowest correlation F_l. Plot the joint distribution for F_h and the target Y. Plot the joint distribution for F_l and the target Y. Compare these joint distribution plots and explain your observations. (To produce the joint distribution density plots, you can use the seaborn library and set the "kind" parameter to "hex" or "kde".)

Task II - Estimating Service Metrics from Device Statistics

1. **Model Training:** Train three models M_1, M_2, M_3 based on the data X' and Y using the methods linear regression, random forest regression, and neural network regression. Train and test your models M_i with the *validation-set technique*: you split the set of observations into two parts, the *training set* for computing the model M_i and the *test set* for evaluating the accuracy of M_i. From the complete set of observations, you select uniformly at random 70% of the observations to form the training set and then assign the remaining 30% to the test set.

2. **Model Accuracy:** Compute the *estimation error* of the models M_i on the test set. We define the estimation error as the *Normalized Mean Absolute Error* $(NMAE) = \frac{1}{\bar{y}}(\frac{1}{m}\sum_{j=1}^{m}|y_j - \hat{y}_j|)$, whereby $\hat{y}_j$ is the model estimation for the measured service metric y_j, and $\bar{y}$ is the average of the observations y_j of the test set. Note that $\hat{y}_j = M_i(x_j)$.

3. **Model Evaluation:** Provide this evaluation of accuracy for all three methods. For neural network regression, describe how you perform a hyper-parameter search and identify effective hyper-parameters that give the best accuracy for your trace.

4. As a baseline for M_i, use a naïve method which relies on Y values only. For each $x \in X$ it predicts a constant value $\bar{y_{tr}}$ which is the mean of the samples y_j in the training set. Apply the naïve method and compute the NMAE for the test set.

5. Choose one method (either linear regression, random forest, or neural network) and produce a time series plot that shows both the measurements and the model estimations for the target on the test set. Show also the prediction of the a naïve method (see Figure 1). For this plot choose a time interval with 1 000 samples of the test set and sort the samples according to time stamps.

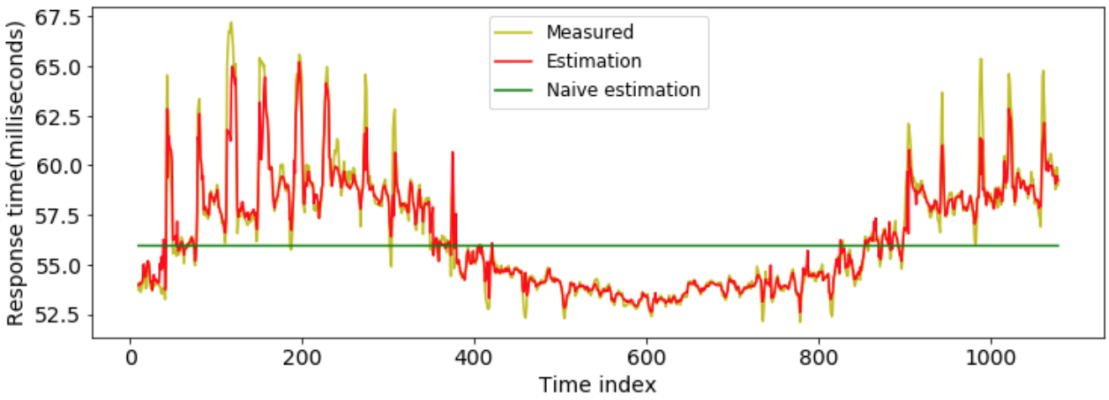

Figure 1: Example: time series plot with measurements and predictions (VOD service)

6. Produce a density plot and a histogram for the target values on the test set. Set the bin size of the histogram to 1 frame for Video Frame Rate or $1ms$ for Response Time.
7. Produce a density plot of the estimation errors $y_j - \hat{y}_j$ from the test set evaluation for all models M_1, M_2, M_3 on the same figure.
8. Based on the above results, figures, and graphs, discuss and compare the accuracy and computational overhead of estimating the target metric for the regression methods you used. You can measure the computational overhead as the training time of the model.

Task III - Studying the Impact of Data Pre-processing and Outlier Removal on the Prediction Accuracy

1. In Task I, you performed six different forms of pre-processing on the collected device measurements X. Together with the unprocessed measurements, you have seven different design matrices $X_0, ..., X_6$.
2. Perform a comparative study to find out which of the pre-processing methods performs best in terms of prediction accuracy. Study this issue for linear regression and random forest regression. Which pre-processing method works well for both regression methods?
3. Detect and remove outliers. Take the design matrix X_5, whose feature columns are standardized. We call a sample an outlier when one of its components has an absolute value larger than a given threshold T. Plot the number of outliers of your data set in function of T within the range [5,120].
   The idea is that once the threshold T is decided, all samples with components whose absolute values are larger than T are removed from the data set.
4. Investigate the error of a regressor in function of T for your data set. Start with the initial data set $S = X_5$ (the design matrix whose feature columns are standardized) and produce reduced data sets $S_1, ..., S_{10}$ for $T = 10, 20, 30, ...100$. For each S_i use linear regression and random forest regression and evaluate the learned models. Assess the effect of outlier removal for both methods by producing a plot that shows the error ($NMAE$) in function of T.
5. (Optional) In the previous step, use 10-fold cross-validation [2]. Present the results in a plot that shows the mean of 10 values and the 95% confidence interval in function of T.

Quiz

Consider the following histogram as the conditional distribution $P(Y|X = x_0)$. For this distribution, what is the expected value $\mathbb{E}[P(Y|X = x_0)]$? What is the most probable value? What is the 90th percentile?

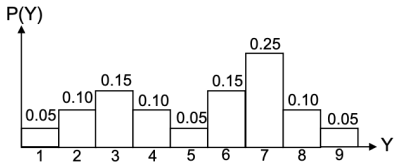

Figure 2: Histogram plot of the $P(Y|X)$ for quiz.

Task IV - Predicting the Distribution of Target Variables using Histograms

1. For this task choose the data set X that has been the outcome of pre-processing and outlier removal. For pre-processing choose the method that gives the best results in Task III. For outlier removal, choose a threshold that keeps 99% of the samples in the data set.

2. The basic idea of this task is to discretize the target space Y and use a histogram estimator for predicting $P(Y|X)$. This means that each $x \in X$ is mapped onto a histogram on Y.

3. In the case of Y representing the Video Frame Rate, the y values are integers, e.g., 15 Frames/sec. Consider the histogram on the interval $y \in [0.5, 30.5]$ with a bin size of 1. This results in 30 bins, with mid points 1, 2, .., 30.

4. In the case of Y representing the Response Time, consider the histogram on the interval $y \in [y_{min}, y_{max}]$ whereby y_{min} is the minimum y value in the training set and y_{max} is the maximum y value. Divide this interval into 25 bins of equal size.

5. Consider each bin of the histogram as a separate class and use a random forest classifier to predict the probability for each class. The predicted value is computed as the expectation based on the probability over the mid point of each bin.

6. To evaluate the accuracy of the method, compute the error of the predicted values with respect to the measured values over the test set. Express the error as $NMAE$. Compare the result with those from Tasks II and III.

7. For illustration purposes, chose two x samples from the test set and draw the two predicted histograms. For both histograms indicate the measured y values.

Optional Task V - Predicting Percentiles of Target Metrics

1. The goal of this task is to use the histogram estimator from Task IV to predict the 25^{th}, 50^{th}, and 95^{th} percentile values of the target Y.

2. Given an instance of a histogram described in Task IV.7 (for Video Frame Rate or Response Time), describe how you compute the above given percentile values.

3. Consider the x samples in your data set that belong to the first hour (3600 second) of the experiment. Produce a time series plot that shows the predicted 25^{th}, 50^{th}, and 95^{th} percentile values of the target Y, together with the measured values.

4. To evaluate the accuracy of the predicted percentile values, you compute $\frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\{y^{(t)} \leq a_{perc}(x^{(t)})\}$, for $perc = 0.25$, 0.5, and 0.95 on the above samples. In the formula, $(y^{(t)}, x^{(t)}), t = 1...n$ are the samples of the test set. $a_{perc}(x^{(t)})$ is the predicted percentile value for $x^{(t)}$. You can obtain this value from the predicted histogram for $x^{(t)}$. $\mathbb{1}\{Q\}$ denotes the indicator function, which takes the value 1 if Q is true, and the value 0 otherwise. The summation is over the test set. The formula gives an estimation for *perc*. (The Glivenko-Cantelli Theorem gives the foundation for this estimation method.)

Optional Task VI - Improving the Prediction Accuracy and Predicting into the Future

This task gives you freedom of choice with respect to the details and the methods of the investigation, as well as the presentation of the results.

In Tasks II to V you have studied predictors that map a measurement value x_t at time t to a predicted service quality value $\hat{y}_t$ at time t. In this task, you investigate to which extent considering more than one measurement leads to a more accurate prediction.

Specifically, you study predicting y_t from $[x_{t-l}, x_{t-l+1}, \ldots, x_{t-1}, x_t]$. You can experiment with different values for l (for instance $3, 5, 10$), different spacing between measurements (for instance $1sec, 2sec, 5sec$), and different prediction methods (for instance linear regression, random forest regression, neural network regression). The goal is to increase the prediction accuracy of $\hat{y}_t$ compared to using $l = 0$ for a given trace.

In addition, you study predictors that predict y_{t+h} from x_t or from $[x_{t-l}, x_{t-l+1}, \ldots, x_{t-1}, x_t]$. Similar to above, you choose values for h and l as well, as the time spacing. The larger the value for h, the longer you predict into the future; the larger the value for l the more measurements you consider from the past.

Select a trace you have been working with, so you can compare the results from this task to those you obtained in earlier tasks.

Resources

Linear regression: You can find the theory of this concept in chapter 5 of [2]. For implementation you can use [3].
Neural Networks regression: You can find theory of this concept in chapter 6 of [2]. For implementation you can use [4].
Data pre-processing and outlier removal: For the theory of the this concept you can read chapter 6 of [5]. For implementation you can use [6].
Predicting the distribution of the target variable and estimating the percentiles: For the theory of these concepts you can use [7] and for the implementation you can use [8].

References

[1] F. S. Samani, H. Zhang, and R. Stadler, "Efficient learning on high-dimensional operational data," in *2019 15th International Conference on Network and Service Management (CNSM)*, IEEE, 2019.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[3] Sklearn Developers, "sklearn.linear_model.linearregression." 2007-2025. [Online]. Available at: `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html`, Accessed on: October 13, 2025.

[4] J. Brownlee, "Regression tutorial with the keras deep learning library in python." 2016. [Online]. Available at: `https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/`, Accessed on: October 13, 2025.

[5] H. Zhang, "Efficient learning on high-dimensional operational data," Master's thesis, KTH, School of Electrical Engineering and Computer Science (EECS), 2019.

[6] Sklearn Developers, "Preprocessing data." 2007-2025. [Online]. Available at: `https://scikit-learn.org/stable/modules/preprocessing.html`, Accessed on: October 13, 2025.

[7] F. S. Samani, R. Stadler, C. Flinta, and A. Johnsson, "Conditional density estimation of service metrics for networked services," *IEEE Transactions on Network and Service Management*, 2021.

[8] Sklearn Developers, "Randomforestclassifier." 2007-2025. [Online]. Available at: `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`, Accessed on: October 13, 2025.