Varun RQ3 Evaluation

Sebastian Hönel

April 28, 2023

Contents

1 Load The Data

```
td.notions <- c("CodeClimate", "Codiga", "SonarQube")
data <- NULL

for (tdn in td.notions) {
  temp <- read.csv(file = paste0("data/", tdn, "_debt_results.csv"), header = TRUE)
  temp$Notion <- tdn
  data <- rbind(data, temp)
}
data$Model_type <- factor(x = data$Model_type, levels = unique(data$Model_type))
data$PCA <- as.logical(data$PCA)
data$Domain <- as.logical(data$Domain)
data$Notion <- factor(x = data$Notion, levels = unique(data$Notion))
temp <- NULL
```

2 Which Notion Can We Predict Best?

We know that the TD notions of Codiga and SonarQube differ (e.g., it is much more common to have a large value for the latter, see Figure 1). This makes them difficult to compare. However, we can transform the corpus' projects' TD into CDFs and *normalize* the predicted values. In other words, the CDF-transformation will give us results that we can compare.

Now we got **four** different flavors per TD notion: With/without Domain and with/without performing PCA on the data. We also obtained $1,000$ RMSEs/MAEs per flavor. If we consider the MAE here, it is the mean absolute difference between the actual and predicted TD. If we take the obtained MAEs and convert them

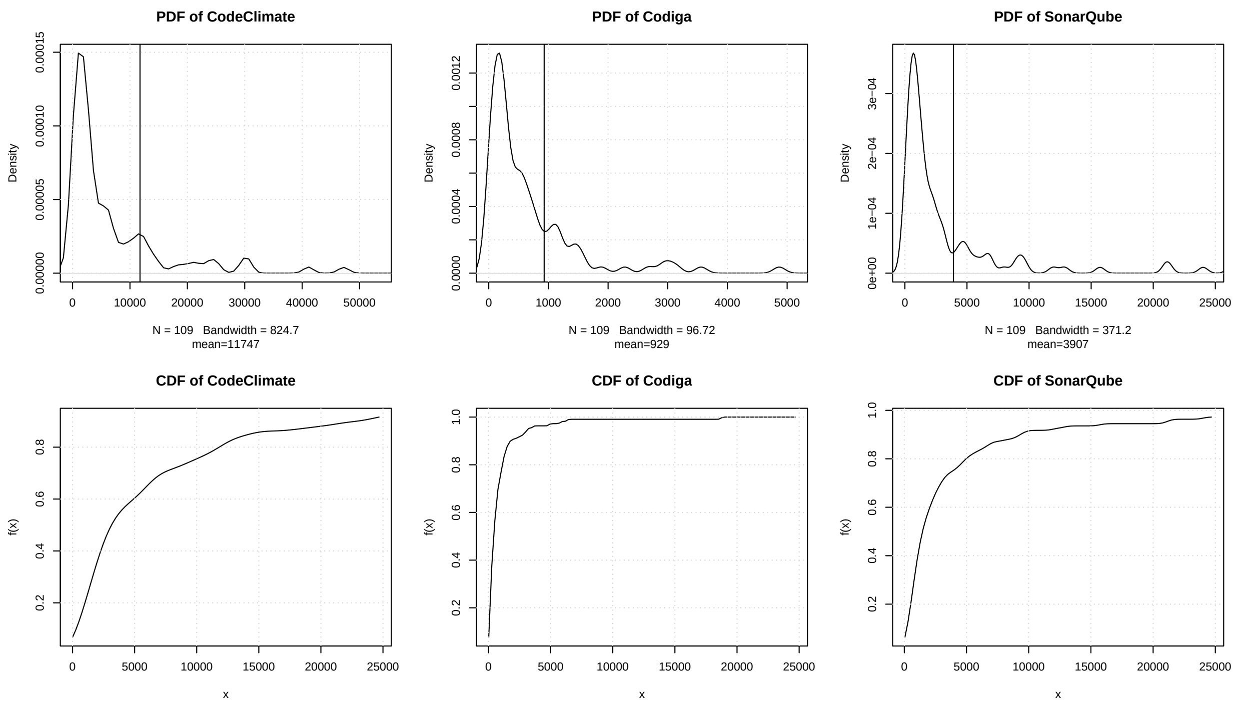

Figure 1: Distribution of technical debt for all three notions in the Qualitas.class corpus.

into probabilities using the corresponding CDFs, we will get a good idea for each flavor of estimator whether it predicts low or high deviations. Using the CDFs also allows us then to compare these results, which would otherwise not have been possible. The last dimensions we need to consider here is the used **model**. The result is shown in Table 1.

```
comparison <- NULL
use.grid <- expand.grid(list(keep_Domain=c(TRUE, FALSE), do_PCA=c(TRUE,FALSE)))

for (tdn in td.notions) {
  for (model_type in levels(data$Model_type)) {
    temp <- `rownames<-`(x = matrix(nrow = 1, ncol = 4), value = paste0(tdn, "_", model_type))

    use.labels <- c()
    for (idx in 1:nrow(use.grid)) {
      row <- use.grid[idx,]
      use.labels <- c(use.labels, paste0(if (row$keep_Domain) "WithDomain" else "NoDomain", "_", if (ro
      temp[1, idx] <- mean(cdfs[[tdn]](
        data[data$Notion == tdn & data$Model_type == model_type & data$Domain == row$keep_Domain & data$
    }

    comparison <- rbind(comparison, `colnames<-`(x = temp, value = use.labels))
  }
}
```

The above results tell us the following: the lower the average value, the better the flavor. The best value for CodeClimate is **XGB**, **including domain**, **not doing PCA** (≈ 0.709). The same flavor is the best for Codiga (≈ 0.577), as well as SonarQube (≈ 0.718).

Table 1: Showing all three notions of TD, grid configurations, and models.

	WithDomain_UsePCA	NoDomain_UsePCA	WithDomain_NoPCA	NoDomain_NoPCA
CodeClimate_RF	0.7606750	0.7660496	0.7452672	0.7519722
CodeClimate_XGB	0.7498625	0.7581327	0.7086790	0.7194362
CodeClimate_MLP	0.9056478	0.9087426	0.9162333	0.9183945
CodeClimate_LR	0.9328974	0.9311435	0.9530537	0.9524264
Codiga_RF	0.6795504	0.6834638	0.6536617	0.6601129
Codiga_XGB	0.6525374	0.6550226	0.5765633	0.5856857
Codiga_MLP	0.8165407	0.8165647	0.8252067	0.8308815
Codiga_LR	0.8253077	0.8213767	0.8698335	0.8683150
SonarQube_RF	0.7396159	0.7412417	0.7301272	0.7341624
SonarQube_XGB	0.7335765	0.7376694	0.7178658	0.7215078
SonarQube_MLP	0.8185164	0.8204274	0.8366887	0.8390546
SonarQube_LR	0.8076266	0.8096504	0.9079272	0.9083246

That also means, Codiga $>$ CodeClimate $\approx$ SonarQube in terms of what we can predict best.

Let's show the distributions of the original and CDF-normalized MAEs (Figure 2):

I also include the test for normality. As you see, none of the residual MAEs are normally distributed, nor do they have a unimodal distribution (the residual MAEs for SonarQube are almost normal though). Therefore, we can only apply Chebyshev's inequality for estimating confidence intervals (Tchébychef 1867).

We really only should use the last row to compare how well we can predict the different notions of technical debt. For CodeClimate and SonarQube, we actually do not get models that achieve scores $>\approx 0.5$.

3 Continuous Confidence Intervals

In Table 2 shows the three-sigma rule, as well as Chebyshev's rule (I only include the three-sigma rule as demonstration, but it is not applicable here). It shows that, for example, only for very low confidences, we expect our champion model to deviate rather slightly from the expected mean MAE. For example, our confidence is only 5% that the predictions for the type of champion model on Codiga's TD notion deviates between 312 and 837 around a mean of 575. In other words, even the champion model is "usually" (on average) of by 575. However, it may be off less (down to 312) or more (up to 837). We can "guarantee" this with a confidence of 5%.

Such a low confidence has no real practical usage. So, if we look at a more reasonable example, say, 90%, the deviation from the expectation (mean) must be more extreme. Note that I limited the models so they can only predict values $>= 0$. So with 90% confidence, the deviation from the actual technical debt, for Codiga, is $<\approx 1,383$.

Recall that a typical value for Codiga's notion of technical debt is 929. In other words, running Codiga on a random project will on average yield a TD of 929. If we're off by more than $1,300$, it could mean that we predict (roughly) that a project has no TD when it actually has TD, or that we predict the TD to be much higher. These are just examples off how "off" we could be with those 90%.

But here is where it also gets interesting. Suppose you are happy with a lower confidence, just to get "ballpark" kind of figures. So if, e.g., 50% confidence is enough, this predictor might become useful. Let's say you are dealing with projects that usually have a very high TD, then being off by $\approx 1,000$ is perhaps acceptable for getting a first quick impression. Remember that this here is a trade-off; evaluating TD is perhaps really expensive in real-life, as you need experts who have to do this qualitatively, evaluating lots of unstructured data (Matsubara et al. 2021).

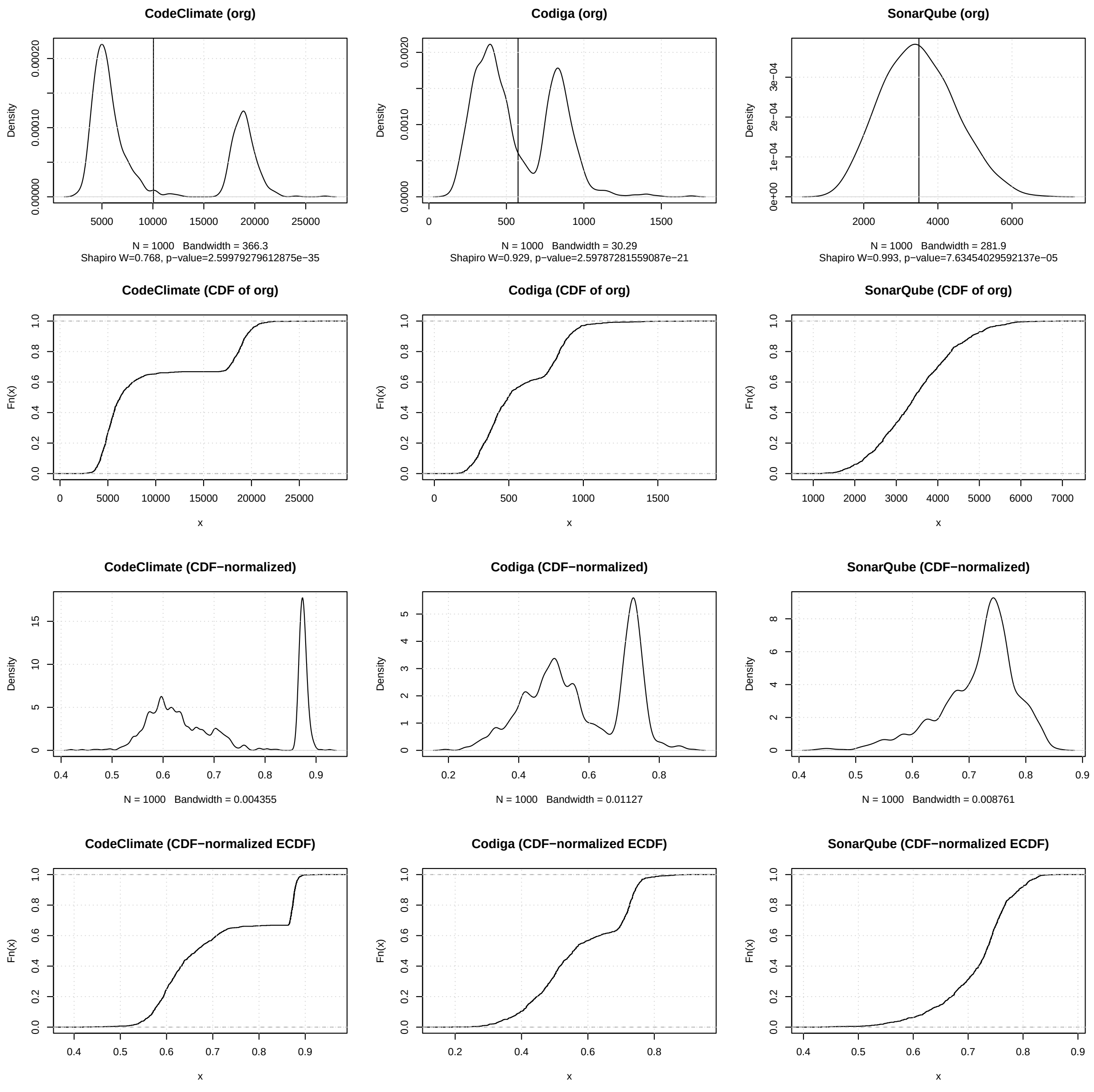

Figure 2: The distributions of MAEs, per notion of TD. Shown is the PDF and CDF of the original MAEs, as well as the PDF and CDF of the CDF-transformed MAEs so that we can compare.

Table 2: Confidence in relation to inequalities.

Confidence	Low	Mean1	High	Chebyshev_Low	Mean2	Chebyshev_High
5.00	559.0113	575.04	591.0778	312.7156	575.04	837.3734
15.00	526.6895	575.04	623.3996	297.7135	575.04	852.3756
25.00	493.5727	575.04	656.5164	279.8031	575.04	870.2860
35.00	459.0236	575.04	691.0654	257.9044	575.04	892.1847
45.00	422.2053	575.04	727.8838	230.2768	575.04	919.8123
50.00	402.5866	575.04	747.5025	213.4491	575.04	936.6400
55.00	381.8951	575.04	768.1940	193.8895	575.04	956.1996
65.00	336.0826	575.04	814.0065	142.8553	575.04	1007.2338
68.27	319.3524	575.04	830.7367	121.1312	575.04	1028.9578
75.00	280.9156	575.04	869.1734	63.6714	575.04	1086.4177
80.00	247.3690	575.04	902.7201	3.3120	575.04	1146.7771
85.00	206.9757	575.04	943.1134	0.0000	575.04	1235.2245
90.00	154.4775	575.04	995.6115	0.0000	575.04	1383.5965
95.00	73.9081	575.04	1076.1810	0.0000	575.04	1718.5097
95.45	63.6708	575.04	1086.4183	0.0000	575.04	1773.7217
97.50	1.9479	575.04	1148.1412	0.0000	575.04	2192.1485
98.00	0.0000	575.04	1169.8605	0.0000	575.04	2383.0217

```r
library(dplyr)
```

```r
temp <- data[data$Notion == "Codiga" & data$Model_type == "XGB" & data$Domain == TRUE & data$PCA == FALS

temp.df <- data.frame(
  Confidence = sort(c(seq(from = 5, to = 95, by = 10), 50, 80, 90, 68.27, 95.45, 97.5, 98))
) %>% group_by(Confidence) %>% summarize(
  Low = max(0, round(qnorm(c(.5 - Confidence / 200), mean = mean(temp), sd = sd(temp)), 4)),
  Mean1 = round(mean(temp), 2),
  High = round(qnorm(c(.5 + Confidence / 200), mean = mean(temp), sd = sd(temp)), 4),
  Chebyshev_Low = max(0, round(mean(temp) - sd(temp) * sqrt(1 / (1 - Confidence / 100)), 4)),
  Mean2 = round(mean(temp), 2),
  Chebyshev_High = round(mean(temp) + sd(temp) * sqrt(1 / (1 - Confidence / 100)), 4),
  .groups = "drop")
```

References

Matsubara, Patrícia Gomes Fernandes, Bruno Freitas Gadelha, Igor Steinmacher, and Tayana Uchôa Conte. 2021. "SEXTAMT: A Systematic Map to Navigate the Wide Seas of Factors Affecting Expert Judgment Software Estimates." *Journal of Systems and Software*, 111148. https://doi.org/10.1016/j.jss.2021.111148.

Tchébychef, Pafnuty. 1867. "Des Valeurs Moyennes (Traduction Du Russe, n. De Khanikof." *Journal de Mathématiques Pures Et Appliquées*, 177–84. http://eudml.org/doc/234989.