Characterizing Anomalies with Explainable Classifiers

Anonymous Author(s)
Affiliation
Address
email

Abstract

As machine learning techniques are increasingly used to make societal-scale decisions, model performance issues stemming from data-drift can result in costly consequences. While methods exist to quantify data-drift, a further classification of drifted points into groups of similarly anomalous points can be helpful for practitioners as a means to combating drift (e.g. by providing context about how/where in the data pipeline shift might be introduced). We show how such characterization is possible by making use of tools from the model explainability literature. We also show how simple rules can be extracted to generate database queries for anomalous data and detect anomalous data in the future.

Keywords: Data-drift, Anomaly Detection, Explainability, SHAP

1 Introduction

Fielded machine learning systems help corporations, banks, and institutions make key decisions on a daily basis. However, these systems are often used and trusted despite the fact that input covariates may be distributed differently than they were during training. Demand-forecasting algorithms may be thrown off by new consumer practices that began during the COVID-19 pandemic [1]. Fraud-detection algorithms may gradually become ineffective as agents that intend to commit fraud develop more sophisticated strategies [2, 3]. While various techniques[1] exist to measure the extent to which the distribution has shifted, such measurements alone do not give practictioners intuition about what characterizes shifted points. These characterizations can be especially difficult to construct and intuit when the correlations between covariates shift as opposed to the marginal distributions: metrics that track univariate drift for each feature are among the most interpretable, but in these cases they would fall short.

We propose a method based on tools from the explainable classification literature to (i) measure the total multivariate covariate shift present, (ii) identify and characterize clusters of similarly anomalous points, and (iii) construct simple rules that practitioners may use to isolate anomalous data for further exploratory data analysis. Our approach fundamentally builds on the idea of discriminative distance [5], wherein the statistical distance between two distributions is thought of in terms of the performance of a classifier that aims to tell the two distributions apart. If the classifier is able to distinguish between the two distributions easily, then we think of the two distributions as being far apart and vice-versa. Our approach takes this idea a step further by making use of the classifier itself (by means of tools from the model explainability literature) to characterize and identify data points that are anomalous for similar reasons. While we evaluate our approach on tabular datasets, in principle, our approach can be implemented on any form of data (e.g. images, text) for which tractable classification algorithms with local additive explanations exist.

[1] See [4] for an in-depth exposition.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

	Shape	G	$\lvert R_g \rvert$	Potentially Shifted Features
DCCC	(30000, 24)	2	1192, 1752	$\overbrace{\texttt{PAY_4}, \texttt{BILL_AMT3}, \texttt{BILL_AMT4}, \texttt{BILL_AMT6}}^{F_1^{num}}, \overbrace{\texttt{PAY_2}, \texttt{PAY_6}}^{F_1^{cat}}, \underbrace{\texttt{BILL_AMT3}, \texttt{PAY_AMT5}, \texttt{BILL_AMT2}, \texttt{BILL_AMT5}}_{F_2^{num}}, \underbrace{\texttt{default}}_{F_2^{cat}}$
Statlog	(1000, 20)	1	25	$\underbrace{\texttt{Dur._in_Cur_Addr.}, \texttt{Type_of_Apt.}, \texttt{No._Dependents}, \texttt{Guarantors}}_{F_1^{cat}}$

Table 1: **Synthetically Drifted Data.** Using the synthetic data generation scheme described in Section 3, we generated drifted test datasets for the *Default on Credit Card Clients* dataset [12, 13] and the *Statlog (German Credit Data)* dataset [13]. The reported shape is the size of the full dataset. To obtain the drifted sets we first randomly split into a train and test set, where the test set comprises 25% of the original data, and then subsequently applied the methods described in Section 3.

2 Related Work

Isolation Forests and Statistical Divergences Isolation forests [6] are a common technique used to identify anomalies in datasets. They function using the underlying intuition that anomalous datapoints are likely to be separated in data-space from the non-anomalous points. However, for detecting anomalies in model covariates, isolation forests may not obtain good performance relative to discriminative approaches as they do not make use of the knowledge model designers have about the explicit partitioning of data as either training or test data. Statistical divergences such as the KL-Divergence are also often used to detect data drift, but due to issues that arise from data sparsity in higher dimensions, they are typically only applied to compute univariate drift.

Discriminative Distance and Domain Classification Our work can be viewed as an extension of discriminative distance/domain classification approaches [5, 7, 4]. These approaches measure the statistical distance between two distributions in terms of the performance of an auxiliary classifier that aims to distinguish between samples from the two distributions. We build on these approaches to further use the learned model to characterize anomalies.

Local Additive Explanations Given a model $f : \mathbb{R}^d \to \mathbb{R}$, a local additive explanation method ϕ assigns to each point $\boldsymbol{x} \in \mathbb{R}^d$ in the dataset and explanation vector $\phi(\boldsymbol{x}) \in \mathbb{R}^d$ that satisfies $\phi_0 + \sum_i \phi(\boldsymbol{x})_i = f(\boldsymbol{x})$ for some constant ϕ_0. The components $\phi(\boldsymbol{x})_i$ are intended to denote the contribution of the value of feature i to the final model prediction. Examples of popular local additive explanation methods include SHAP [8] and LIME [9]. We use SHAP in our approach to identify similarly anomalous groups of points.

High-Dimensional Visualization To visualize and qualitatively understand the performance of our approach, we make use of nonlinear dimensionality reduction tools to project high-dimensional point clouds into two dimensions. We principally make use of an approach known as UMAP [10], which assumes a Riemannian structure on the data manifold to make good visualizations. Other popular tools such as TSNE [11] exist for this task as well.

3 Synthetic Data Generation

To construct a synthetically drifted test set from some original test set with numerical columns $N \in \mathbb{R}^{m \times n}$ and categorical columns $C \in \texttt{Object}^{m \times c}$, we first randomly select a number G to determine the number of anomalous groups we will construct. For each $g \in [G]$, we randomly select some disjoint subset $R_g \subset [m]$ of rows to belong to group g, and subsequently randomly select a (small) subset of numerical features $F_g^{num} \subset [n]$ of size s_{num} and a similar subset of categorical features $F_g^{cat} \subset [c]$ of size s_{cat} to perturb for each element in the group.

To build a rich class of possible numerical data drifts, we first generate a random group-specific affine transformation $T_g^{num} : \mathbb{R}^{s_{num}} \to \mathbb{R}^{s_{num}}$, and repeatedly apply it to build the drifted numerical columns $N' \in \mathbb{R}^{m \times n}$. Formally, for some $k \in \mathbb{N}$, we let $N'[R_g, F_g^{num}] := \underbrace{T_g^{num} \circ \cdots \circ T_g^{num}}_{k \text{ times}} \left(N[R_g, F_g^{num}]\right)$ and subsequently let the remainder of N' be unchanged

(i.e. $N'[R_g, \overline{F_g^{num}}] := N[R_g, \overline{F_g^{num}}]$, $N'\left[\bigcap_{g\in[G]} \overline{R_g}, :\right] := N\left[\bigcap_{g\in[G]} \overline{R_g}, :\right]$). We define $T_g^{num}(Z) := (I_{s_{num}\times s_{num}} + \epsilon_g)Z + S_g$ where $\epsilon_g \in \mathbb{R}^{s_{num}\times s_{num}}$ and $S_g \in \mathbb{R}^{s_{num}\times s_{num}}$ are randomly generated Gaussian noise.

To shift the categorical features, we first select an element α_i at random from the respective support of each column $i \in F_g^{cat}$ to be shifted. We then "lock" the feature values for each row in group g to be the corresponding α_i. Formally, we let the shifted categorical columns be given by $C' \in \texttt{Object}^{m\times c}$, and let $C'[R_g, F_g^{cat}] := \begin{bmatrix} \alpha_{i_1} & \dots & \alpha_{i_{s_{cat}}} \\ \vdots & \ddots & \vdots \\ \alpha_{i_1} & \dots & \alpha_{i_{s_{cat}}} \end{bmatrix}$. As before, we let the remainder of C' be unchanged from C: $C'[R_g, \overline{F_g^{cat}}] := C[R_g, \overline{F_g^{cat}}]$ and $C'\left[\bigcap_{g\in[G]} \overline{R_g}, :\right] := C\left[\bigcap_{g\in[G]} \overline{R_g}, :\right]$.

Running Examples We use our data generator to generate synthetically drifted test data for two datasets, which we use as running examples for the remainder of the paper. Details regarding the datasets and the anomalous groups generated can be found in Table 1.

4 Identifying and Characterizing Anomalies

We now illustrate the function of our method on our two running examples. In the *Default on Credit Card Clients* example, the underlying dataset and the two anomalous groups are relatively larger (constituting a net $\approx 30\%$ of the test dataset), and they have been perturbed by a greater extent. In the *Statlog* example, the underlying dataset is 30 times smaller, and the anomalous data makes up just 10% of the test data. Ultimately, we will show that in both cases, it is indeed possible to approximately recover all of the information contained in Table 1 given just the training and drifted test set alone.

Measuring Drift We first describe how to quantify the net distributional shift that has occurred in a way that takes into account changes not only in marginal distributions, but also in correlations. While there are a few well-known approaches to do this (see Section 2), we take a discriminative distance approach. More specifically, we train a random forest classifier f to distinguish between training and test instances. We visualize the train and test sets, as well as the predicted score generated by the random forest on a holdout set.

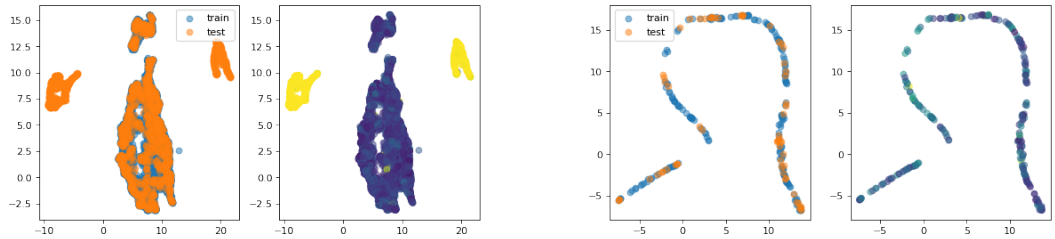


Figure 1: **UMAP Visualizations of Data Drift.** On the left panel of each of the two figures, we plot the true train and (synthetic) test distributions, and on the right panel we color-code points in the classifier's holdout set by the predicted score (a brighter color implies a higher likelihood of being anomalous). The left figure corresponds to the *Default on Credit Card Clients* example and the right figure corresponds to the *Statlog* example. The holdout AUCs were approximately 0.7 and 0.5 for the *Default on Credit Card Clients* and *Statlog* examples respectively.

In the *Default on Credit Card Clients* example, the two anomalous groups can be clearly seen, and indeed the random forest is able to clearly identify those points as anomalous. While it is unclear exactly where the anomalous points are located in the *Statlog* example, the random forest score does seem to increase in regions where more orange test points are visible. We propose to measure the distributional shift through the AUC[2] of f on a holdout set. If the AUC is close to 0.5, then the

[2]By AUC, we refer to the area under the receiver operating characteristic curve. We direct the reader to [14] for a comprehensive introduction.

net shift is negligible, and if the AUC is close to 1, then the shift is extreme. Indeed, the AUCs as reported in Figure 1 correspond with our intuition regarding the extent of the shift.

Identifying Anomalous Groups We now show how with a local additive explanation method, we can use the classifier f to approximately recover the information contained in Table 1. We make use of the SHAP package [8, 15] to generate these explanations for our random forest classifier. When identifying groups of points that are similarly anomalous, it is important to note that such points may be located far from each other in data-space. For example, if a group of Americans and Europeans adopt the same fraud strategy, this anomalous group might be represented in data-space as two blobs that are spaced apart due to differences in location. Indeed, in the *Statlog* example, we know that 25 points were shifted in the same way, but their locations are not concentrated together. We can already make some insights about the features that were shifted by looking at the SHAP summary plots (see Figure 4 in Appendix A).

To identify groups of similarly anomalous points, we make use of local additive explanations. For each point in data-space, we get a corresponding vector of explanations with the same dimensionality, such that the sum of the explanations is equal to predicted score (as returned by f). If two points are close in explanation-space, we know that their predicted anomaly score is similar, since the sum of the components must be similar. Furthermore, the reason why the points are similarly anomalous must also similar, since the explanation vectors are similar. Thus, by clustering points in explanation-space, we can identify the anomalous groups! We use DBSCAN [16] for clustering, and use Kneedle [17] for hyperparameter tuning as first introduced in [18] (see Figure 5 for details). Clustering is done in high-dimensional space – not in the UMAP visualization. See Figure 2 for a description of the anomaly scores over explanation space.

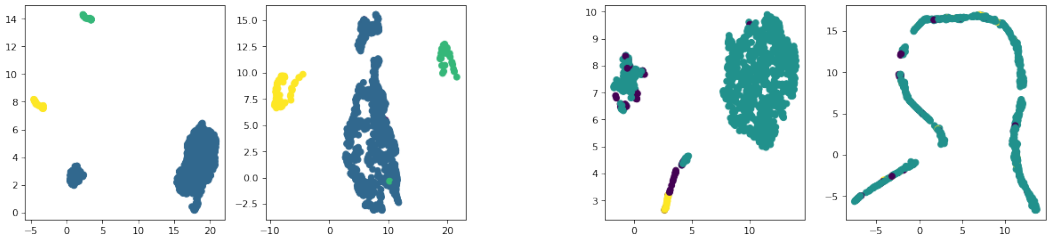


Figure 2: **Clustering in Explanation Space.** The left and right sides of each figure are given by explanation-space and data-space respectively. The left figure corresponds to the *Default on Credit Card Clients* example and the right figure corresponds to the *Stalog* example. Colors correspond to clusters identified by DBSCAN.

By looking at the SHAP signatures of the identified anomalous blobs, we can approximately recover F_g^{num} and F_g^{cat}. From the clustering, we estimate the sizes of the anomalous DCCC clusters to be 960 and 1560, and the size of the *Statlog* cluster to be 22. See Figure 7 for more details.

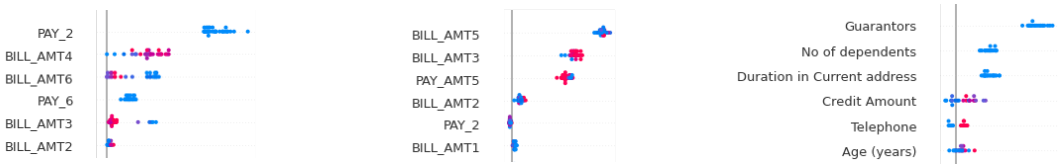


Figure 3: **SHAP Signatures for Anomalous Clusters.** The top SHAP features for cluster 1 (left) and cluster 2 (middle) of the *Default on Credit Card Clients* example, as well as cluster 1 (right) for the *Stalog* example are shown. Notice the overlap between these and the features from Table 1.

Building Simple Rules Finally, using the Skope-Rules package [19], we can extract simple rules that approximately isolate anomalous clusters. These rules can be used as database queries that practitioners can use for exploratory data analysis, or as a means for detecting future anomalous instances. See Figure 8 for more information.

References

[1] Abhinav Garg, Naman Shukla, Lavanya Marla, and Sriram Somanchi. Distribution shift in airline customer behavior during covid-19. *arXiv preprint arXiv:2111.14938*, 2021.

[2] Yvan Lucas, Pierre-Edouard Portier, Léa Laporte, Sylvie Calabretto, Liyun He-Guelton, Frederic Oblé, and Michael Granitzer. Dataset shift quantification for credit card fraud detection. In *2019 IEEE second international conference on artificial intelligence and knowledge engineering (AIKE)*, pages 97–100. IEEE, 2019.

[3] Yvan Lucas and Johannes Jurgovsky. Credit card fraud detection using machine learning: A survey. *arXiv preprint arXiv:2010.06479*, 2020.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[5] Yang Mu, Wei Ding, and Dacheng Tao. Local discriminative distance metrics ensemble learning. *Pattern Recognition*, 46(8):2337–2349, 2013.

[6] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[7] Ziyi Yang, Iman Soltani Bozchalooi, and Eric Darve. Anomaly detection with domain adaptation. *arXiv preprint arXiv:2006.03689*, 2020.

[8] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[9] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[10] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[11] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[12] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2):2473–2480, 2009.

[13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[14] Tianbao Yang and Yiming Ying. Auc maximization in the era of big data and ai: A survey. *ACM Computing Surveys (CSUR)*, 2022.

[15] Scott M Lundberg and Su-In Lee. Consistent feature attribution for tree ensembles. *arXiv preprint arXiv:1706.06060*, 2017.

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[17] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE, 2011.

[18] Nadia Rahmah and Imas Sukaesih Sitanggang. Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra. In *IOP conference series: earth and environmental science*, volume 31, page 012012. IOP Publishing, 2016.

[19] Skope-rules, 2019.

A Additional Figures

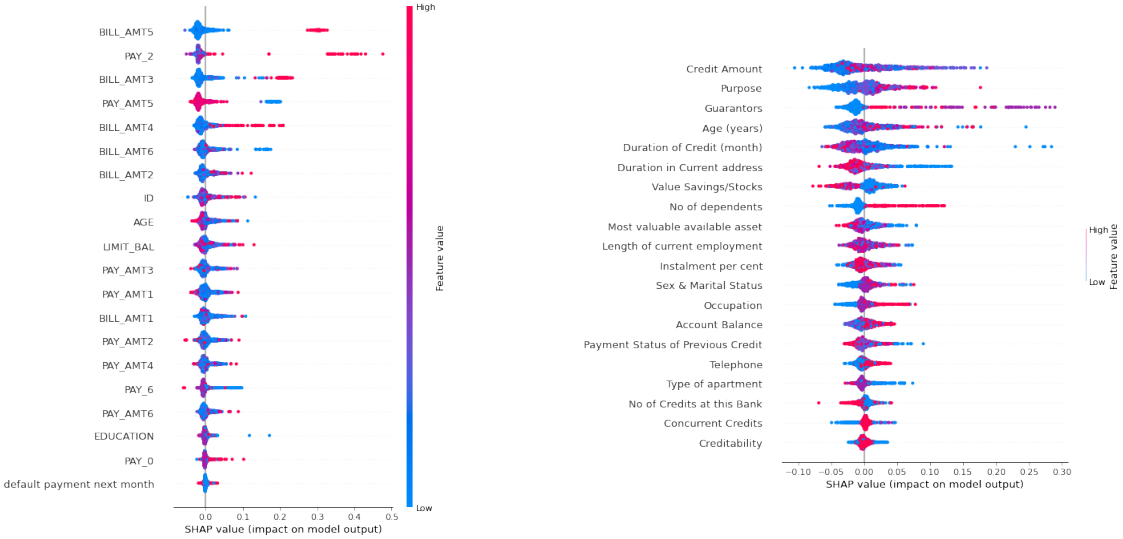


Figure 4: **SHAP Summary Plots.** Mean absolute feature importance is plotted for each feature in the (left) *Default on Credit Card Clients* and (right) *Statlog* examples. The top few features are a combination of those that have the highest variation, and those that are actually shifted (as given in Table 1).

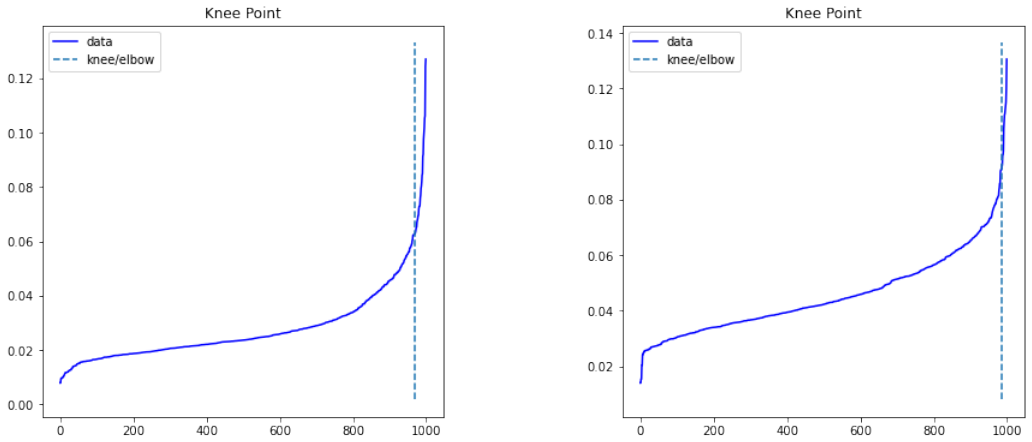


Figure 5: **Kneedle for Hyperparameter Tuning.** To tune the ϵ parameter in DBSCAN [16], we first set the `min_samples` parameter to a reasonable value, and subsequently sort the distances from each point to its `min_samples` nearest neighbor. We then find ϵ by finding the "knee" of this sorted plot, as first suggested in [18].

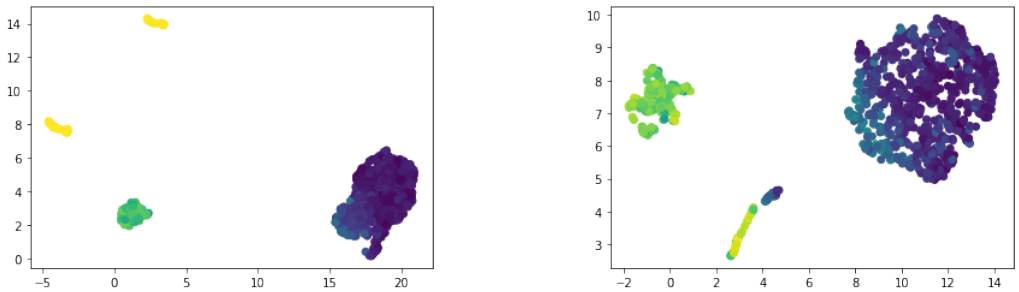


Figure 6: **Anomaly Scores in Expalanation Space.** Points in explanation space are color-coded by the anomaly score as predicted by f in the (left) *Default on Credit Card Clients* and (right) *Statlog* examples. A brighter color corresponds to a more anomalous score. Points tend to disperse into a highly non-anomalous cluster (in blue), a medium anomalous cluster (in green), and highly anomalous cluster(s) (in yellow).

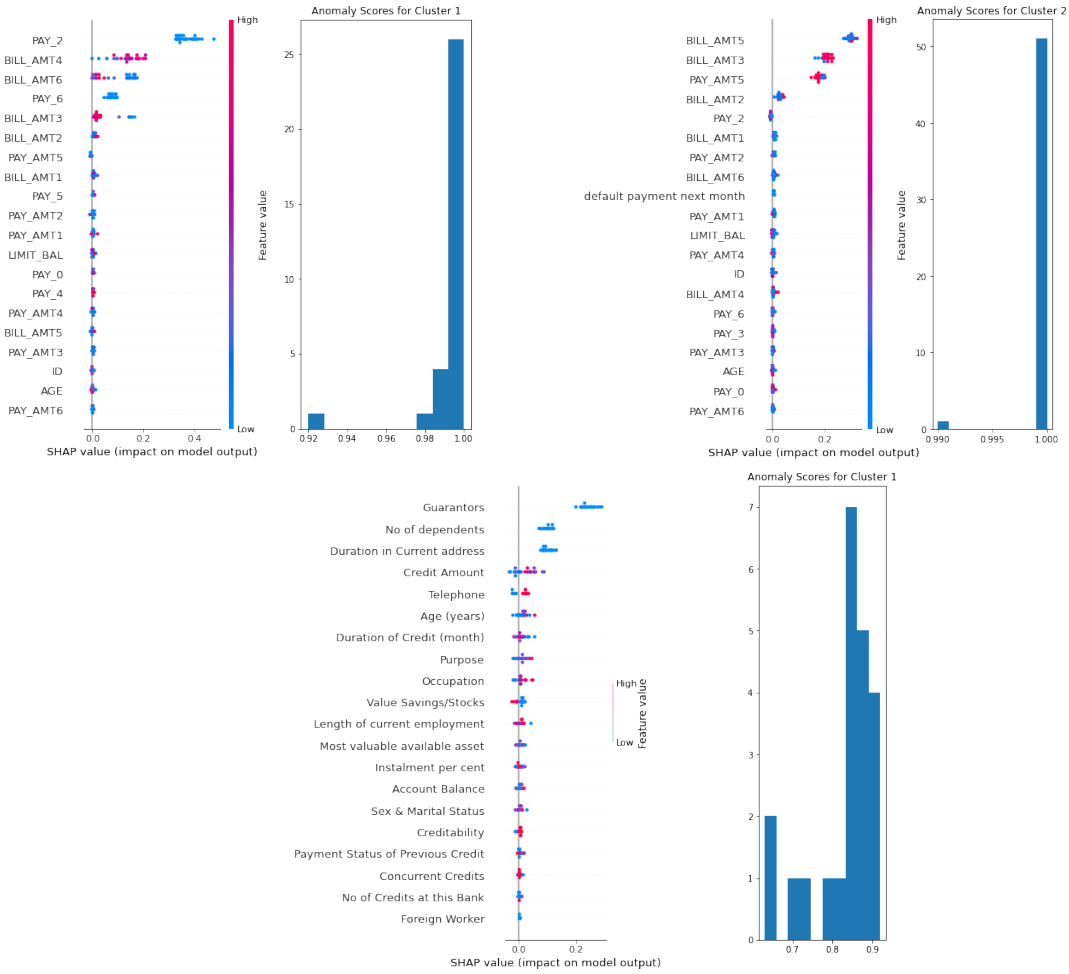
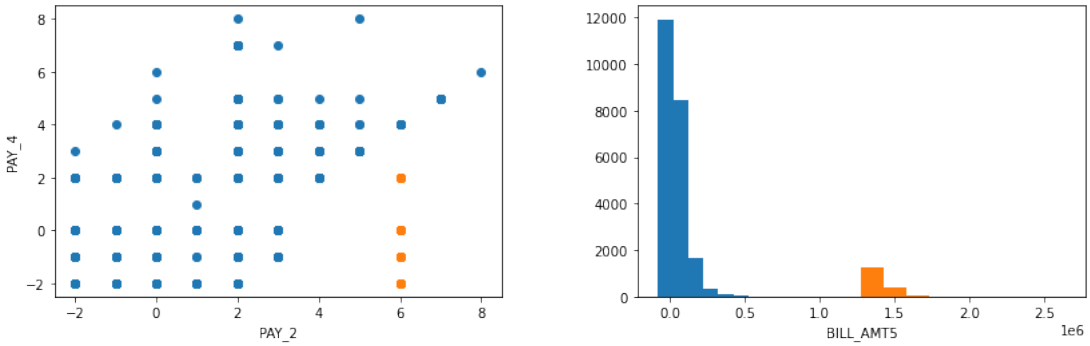


Figure 7: **SHAP Signatures and Anomaly Scores for Anomalous Clusters.** The top SHAP features for cluster 1 (left) and cluster 2 (middle) of the *Default on Credit Card Clients* example, as well as cluster 1 (right) for the *Stalog* example are shown. We also plot a histogram of the anomaly scores among points in the anomalous clusters. The classifier f does well at identifying these clusters as highly anomalous. Notice the overlap between these and the features from Table 1.



Cluster	Rule	Precision/Recall
DCCC 1	$\texttt{PAY_2} > 5.0$ and $\texttt{PAY_4} \leq 2.5$	$1/1$
DCCC 2	$\texttt{BILL_AMT5} > 884308.0$	$1/1$
Statlog 1	$\texttt{Guarantors} > 1.5$ and $\texttt{Dur._in_Cur_Addr.} \leq 1.5$	$0.61/1$

Figure 8: **Skope Rules.** To build simple rules that isolate anomalous clusters, we use the Skope-Rules package [19] to fit an ensemble of decision trees that aim to classify each anomalous cluster. The number of splits per decision tree is limited, so that the final rule depends only on a few features. The top two plots visualize (in orange) the test data satisfying each generated rule for the anomalous clusters in the *Default on Credit Card Clients* example.