A Datasets

A.1 Real Datasets

In this work, we use a total of 9 continuous and 3 categorical datasets, with their dimensions and data types shown in Table 2. The Gas, Covid, and Energy datasets are the same as those used in [9], and we apply identical preprocessing procedures. The Musk2 dataset provides information on musk and non-musk molecules. The Scene dataset describes image characteristics. The MNIST dataset consists of images depicting handwritten digits. The Dilbert dataset is an image recognition dataset of pictures of objects rotated from various orientations. We additionally include 3 internal datasets of biomedical data: Phenotypes, Canine, and Founders datasets. The Phenotypes dataset contains a subset of categorical phenotypes from the UK Biobank (UKB), following the same pre-processing as in [71]. We use the UKB under the application number *(XXXXX-hidden-for-submission)*. The Canine and Founders datasets comprise binary-coded sequences of DNA including Single Nucleotide Polymorphisms (SNPs), representing data for multiple dog breeds and human populations, respectively.

The Polynomial and Cosine are two internally generated datasets consisting of values obtained from deterministic simulations. The Polynomial dataset includes samples where each feature value is obtained by evaluating a second-degree polynomial function $f(x) = ax^2 + bx + c$. The parameters a, b, and c are fixed for each sample. The feature values are derived by evaluating the polynomial function for the different x values within a sample. The values for x, a, b, and c are uniformly sampled from the range of $[-10, 10]$. The Cosine dataset consists of samples with features following a cosine function $f(x) = a \cdot \cos(bx + c)$. Similar to the Polynomial dataset, a, b, and c are fixed for each sample, and each feature value is obtained by evaluating the cosine function for the different x values within a sample. Here, the values of b and c are uniformly sampled within the range of $[-\pi, \pi]$.

Table 2: Datasets used to evaluate DataFix.

Dataset	No. of attributes	No. of samples	Data Type
Gas	8	12,815	Continuous
Covid	10	9,889	Continuous
Energy	26	19,735	Continuous
Musk2	166	6,598	Continuous
Scene	294	2,407	Continuous
MNIST	784	70,000	Continuous
Phenotypes	1,227	31,424	Categorical
Dilbert	2,000	10,000	Continuous
Founders	10,000	4,144	Categorical
Canine	198,473	1,444	Categorical
Cosine	1,000	10,000	Continuous
Polynomial	1,000	10,000	Continuous

A.2 Simulated Probabilistic Datasets

We generate 15 simulated datasets containing 1000 features and 5000 samples by sampling from pre-defined probabilistic distributions, including multivariate Gaussians, with and without transformations, multivariate Bernoulli distributions, Gaussian mixture models, and Bernoulli mixture models. A total of 200 features are shifted in each dataset such that $|C| = 200$ and $|\overline{C}| = 800$. Table 3 describes the distributions used in each dataset. For every dataset, two distributions, p and q, are defined, so that $D(p, q) > 0$ but $D(p_{\overline{C}}, q_{\overline{C}}) = 0$. In fact, for all datasets except for dataset 8, we also have that $D(p_C, q_C) > 0$. A further discussion of the effect of shifts with $D(p_C, q_C) > 0$ and with $D(p_C, q_C) = 0$, and its relation to the equivalence of feature shift localization and feature selection, is provided in the following sections. Because we have access to p and q, we can compute the real divergence between the distributions. In practice, we make use of a Monte Carlo estimate, as shown in figures 5, 18 and 19, because the divergences might not have a closed-form solution or can be computationally intractable.

Datasets 1-3 are based on a multivariate Gaussian, with diagonal covariance used in datasets 1 and 2, and a covariance Σ used in dataset 3. The covariance matrix Σ is defined by a Gaussian kernel such that the ij component is $\Sigma_{ij} = \exp \frac{-||i-j||^2}{s}$, where s acts as a scale parameter, and $0 \leq \Sigma_{ij} \leq 1$ with $\Sigma_{ii} = 1$. In practice, when constructing the covariance matrix, we perform a shuffle of the feature order to better depict tabular data, where, in many cases, the correlation between features does not follow any specific ordering (opposed to images or audio). We use $s = 0.05$ to define Σ_{ij} in dataset 3. Datasets 4 and 5 follow a lognormal distribution, defined as $X = \exp(V)$ with $V \sim \mathcal{N}(\mu, \Sigma)$ and $X \sim \text{Lognormal}(\mu, \Sigma)$. We use $s = 0.05$ and $s = 0.002$ to define Σ_{ij} in datasets 4 and 5, respectively. Datasets 6-8 follow a logit-normal distribution defined as $X = \sigma(V)$ with $V \sim \mathcal{N}(\mu, \Sigma)$ and $X \sim P(\mathcal{N}(\mu, \Sigma))$, where σ is the sigmoid transformation. We use $s = 0.05$, $s = 0.002$, and $s = 0.002$ to define Σ_{ij} in datasets 6, 7, and 8, respectively. Datasets 9-12 follow a multivariate Bernoulli with independent features. Each feature i has a frequency of f_i, where $f \sim P(\mathcal{N}(0, 2I))$, $\epsilon \sim \mathcal{N}(0, I)$, and $(\cdot)_{0,1} = \text{clamp}_{0,1}(\cdot) = \min(\max(\cdot, 0), 1)$ is the clamping function to ensure that the frequencies are between 0 and 1. Dataset 13 follows a Gaussian Mixture Model distribution, with 3 mixtures of equal weights, $\mu_i \sim \mathcal{N}(0, 0.01I)$, and Σ_i defined with $s = 0.3$. Datasets 14 and 15 follow a Bernoulli Mixture Model distribution with 3 mixtures and $f_i \sim \text{Uniform}^d(0, 1)$.

Datasets 1, 3, 4, 6, 9, 10, 11, 12, 13, and 15 apply a shift to the marginal means, such that for all $i \in C$, $\mathbb{E}[p_{C_i}] \neq \mathbb{E}[q_{C_i}]$. Such datasets include marginal shifts of a similar nature as the shifts generated by manipulation types 1, 2, and 6 applied to real datasets. Dataset 2 performs a shift of the marginal standard deviation while maintaining their mean such that for all $i \in C$, $\mathbb{E}[p_{C_i}] = \mathbb{E}[q_{C_i}]$ but $p_{C_i} \neq q_{C_i}$ and $\text{Var}[p_{C_i}] \neq \text{Var}[q_{C_i}]$, leading to a marginal shift similar to the one applied by manipulation type 4 used in the real datasets. Datasets 13 and 15 apply a shift to the mean of just one mixture of the mixture model, leading to only $1/3$ of the samples being shifted, while still ensuring that $\mathbb{E}[p_{C_i}] \neq \mathbb{E}[q_{C_i}]$. Datasets 5, 7, and 14 apply a distribution shift consisting of removing the correlation between features, equivalently to manipulation type 3 applied in real datasets, such that $q_C = \prod_{i \in C} q_i$ and $p_C \neq q_C$. Dataset 8 also applies a shift originating from modifying the correlation between features, but in this case, Σ' is defined as:

$$\Sigma'_{ij} = \begin{cases} \Sigma_{ij} & \text{if } i, j \in C, \text{ or } i, j \in \overline{C} \\ 0 & \text{if } i \in C \text{ with } j \in \overline{C}, \text{ or } j \in C \text{ with } i \in \overline{C} \end{cases} \tag{10}$$

where the correlation of the features within C and within $\overline{C}$ are maintained, but the cross-correlations between the C and $\overline{C}$ are lost, which leads to a shift equivalent to manipulation type 8 applied in real datasets, such that $p_C = q_C$, $p_{\overline{C}} = q_{\overline{C}}$, but $p \neq q$.

B Feature Selection and Feature Shift Localization Equivalence

Section 4 provides a description of when the problem of feature selection and of feature shift localization are equivalent. Namely, when the number of manipulated features is known $|C| = k$, and the divergence of the corrupted features is larger than 0, such that $D_{TV}(p, q) = D_{TV}(p_C, q_C) = 1$, and $D_{TV}(p_{\overline{C}}, q_{\overline{C}}) = 0$, then both problems are equivalent:

$$C = \underset{|C| \leq k}{\text{argmax}}\, I(z_C; t) = \underset{D_{TV}(p_{\overline{C}}, q_{\overline{C}}) = 0}{\text{argmin}} |C| \tag{11}$$

Even if $|C|$ is unknown, if only marginal shifts are present, one can perform feature shift detection by iteratively solving Eq.3 with $k = 1$, and removing the detected features at each iteration from z as long as $I(z_C; t) > 0$, making the iterative feature selection and feature shift localization equivalent problems:

$$C_i = \underset{|C|=1}{\text{argmax}}\, I(z_C; t) \tag{12}$$

Table 3: Probabilistic datasets used to evaluate DataFix.

ID	p_C	q_C	Description
1	$\mathcal{N}(0,\, I)$	$\mathcal{N}(0.5I,\, I)$	Multivariate Gaussians with diagonal covariance and a shifted mean.
2	$\mathcal{N}(0,\, I)$	$\mathcal{N}(0,\, 1.5I)$	Multivariate Gaussians with diagonal covariance and a shifted scale.
3	$\mathcal{N}(0,\, \Sigma)$	$\mathcal{N}(0.5I,\, \Sigma)$	Multivariate Gaussians with non-diagonal covariance and a shifted mean.
4	Lognormal$(0,\, \Sigma)$	Lognormal$(0.5I,\, \Sigma)$	Multivariate lognormal with non-diagonal covariance and a shifted mean.
5	Lognormal$(0,\, \Sigma)$	Lognormal$(0,\, I)$	Multivariate lognormal with non-diagonal (p_C) and diagonal (q_C) covariance.
6	$P(\mathcal{N}(0,\, \Sigma))$	$P(\mathcal{N}(0.5I,\, \Sigma))$	Multivariate logit-normal with non-diagonal covariance and a shifted mean.
7	$P(\mathcal{N}(0,\, \Sigma))$	$P(\mathcal{N}(0,\, I))$	Multivariate logit-normal with non-diagonal (p_C) and diagonal (q_C) covariance.
8	$P(\mathcal{N}(0,\, \Sigma))$	$P(\mathcal{N}(0,\, \Sigma'))$	Multivariate logit-normal with different non-diagonal covariances.
9	Bernoulli(f)	Bernoulli$((f + 0.05\epsilon)_{0,1})$	Multivariate independent Bernoulli with a shifted mean.
10	Bernoulli(f)	Bernoulli$((f + 0.1\epsilon)_{0,1})$	Multivariate independent Bernoulli with a shifted mean.
11	Bernoulli(f)	Bernoulli$((f + 0.5\epsilon)_{0,1})$	Multivariate independent Bernoulli with a shifted mean.
12	Bernoulli(f)	Bernoulli$((f + 1.0\epsilon)_{0,1})$	Multivariate independent Bernoulli with a shifted mean.
13	$\frac{1}{3}\sum_{i=1}^{3}\mathcal{N}(\mu_i,\, \Sigma_i)$	$\frac{1}{3}\sum_{i=1}^{3}\mathcal{N}(\mu'_i,\, \Sigma_i)$	Gaussian Mixture Model with one mixture shifted such that $\mu'_1 = \mu_1 + 10$, $\mu'_2 = \mu_2$, and $\mu'_3 = \mu_3$.
14	BMM$([f_1, f_2, f_3])$	BMM$([f', f', f'])$	Bernoulli Mixture Model with different means $f' = \frac{f_1+f_2+f_3}{3}$.
15	BMM$([f_1, f_2, f_3])$	BMM$([(f_1 + 0.2)_{0,1}, f_2, f_3])$	Bernoulli Mixture Model with one mixture shifted.

where $C = \bigcup_{i=1}^{l} C_i$. In fact, the approach presented in section 4, DF-Locate, can be seen as an approximation of this iterative process, where one or more features are selected at each step by the feature removal policy function.

However, the equivalence of both tasks breaks down for distribution shifts such as the one applied in the manipulation type 8 for real datasets, and in dataset 8 of probabilistic simulations, where $p_C = q_C$ and $p_{\overline{C}} = q_{\overline{C}}$, but $p \neq q$. That is, when considering only the corrupted features C or the non-corrupted features $\overline{C}$ in isolation, the shift is impossible to detect unless all features are considered jointly. Therefore, any feature selection technique that approximates either implicitly or explicitly equation 3, will need a subset of features G containing features from both C and $\overline{C}$ in order to obtain $I(z_G; t) > 0$, because $I(z_C; t) = 0$ and $I(z_{\overline{C}}; t) = 0$. Furthermore, if $|C| = |\overline{C}|$ the problem of feature shift detection becomes unsolvable. This is because, even if a technique was able to properly identify a subset of features $G = C$, it would not be possible to know if the detected subset G contains corrupted or non-corrupted features, that is if $G = C$ or $G = \overline{C}$, making the assumption of $|C| < |\overline{C}|$ a necessary condition.

Figure 3 (right) and Figure 12 show the F-1 score for each manipulation type, indicating that DataFix is able to correctly localize manipulated features with manipulation type 8 on real datasets, despite breaking the equivalence between feature selection and feature shift localization approaches. In contrast, Figure 13, which illustrates the average F-1 on the probabilistic datasets, shows that dataset 8 is the one providing the lowest F-1 scores. While the low F-1 score in dataset 8 can be partly caused by the mismatch between feature selection and feature shift localization problems, it can also

originate from the difficulty of detecting shifts caused by mismatching correlations, as it provides similar performance as in datasets 5 and 7, where correlation shifts (with $p_C \neq q_C$) are applied.

C Feature Shift from Imputation Methods

Imputation and supervised methods trained to reduce the expected mean square error (MSE) between the predicted $\hat{x}_C$ and real x_C features of a given sample x can lead to distribution shifts. Note that the optimal function minimizing $\mathbb{E}_{x\sim P}[||x_C - f(x_{\overline{C}})||^2]$ is the expected value of x_C conditioned in $x_{\overline{C}}$, that is $f^*(x_{\overline{C}}) = \mathbb{E}[x_C|x_{\overline{C}}]$. Therefore, the method that predicts the corrupted (or missing) features C given the non-corrupted (non-missing) features $\overline{C}$, which provides the lowest MSE, will generate predicted samples $\hat{x}_C = f^*(x_{\overline{C}})$, with a distribution where $\mathbb{P}(\hat{x}_C) = 1$ for $\hat{x}_C = \mathbb{E}[x_C|x_{\overline{C}}]$, and $\mathbb{P}(\hat{x}_C) = 0$ everywhere else. If $\text{Var}[x_C|x_{\overline{C}}] > 0$, then $D(\mathbb{P}(\hat{x}_C), \mathbb{P}(x_C)) > 0$, because $\text{Var}[\hat{x}_C|x_{\overline{C}}] = 0$.

For example, given a dataset of samples $x_1, x_2, ..., x_N$, with $x_{i\overline{C}} = x_{j\overline{C}}$ and $x_{iC} \neq x_{jC}$ for all i and j – in other words, $\text{Var}[x_C|x_{\overline{C}}] > 0$ –, the optimal regression model, in terms of MSE, will predict $\hat{x}_{iC} = \mathbb{E}[x_{iC}|x_{i\overline{C}}]$ for all i, such that $\text{Var}[\hat{x}_C|x_{\overline{C}}] = 0$. Similarly, consider a dataset where the features C and $\overline{C}$ are independent such that $p = p_C p_{\overline{C}}$, and $p_C = \mathcal{N}(\mu, I)$, then $\hat{x}_C = \mathbb{E}[x_C|x_{\overline{C}}] = \mathbb{E}[x_C] = \mu$, where $\mu \in \mathbb{R}^{|C|}$ is a constant vector. The simulated probabilistic dataset 1 follows this form, with $p_C = \mathcal{N}(0, I)$ and $q_C = \mathcal{N}(0.5I, I)$, where the method providing the lowest MSE will be the one predicting $\hat{x}_C = \mathbb{E}[x_C|x_{\overline{C}}] = \mu = 0$ for all samples. Figure 6 shows the histogram of the first feature values before and after performing feature correction with multiple techniques. Most techniques, especially imputation-based techniques, output the mean or values highly close to it, which while producing minimum MSE, does not reflect the real distribution, removing or reducing its variance and leading to a distribution shift.

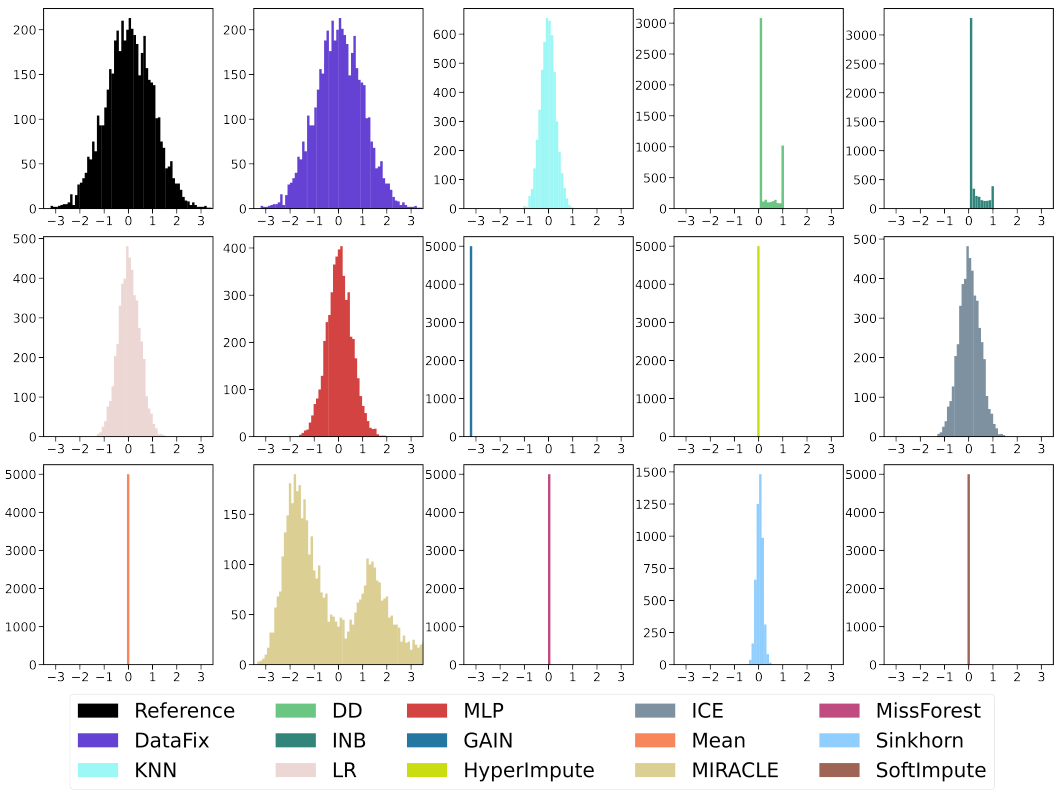

Figure 6: Distribution of first feature values before and after shift correction with various techniques for probabilistic dataset 1.

D DF-Locate

Algorithm 1 DF-Locate: Feature Shift Localization

Inputs:
  X; ▷ Reference
  Y; ▷ Query
  τ; ▷ Feature Selection Threshold
  ϵ; ▷ Divergence Threshold
$X^{(0)} = X$
$Y^{(0)} = Y$
$i = 0$
$k^{(0)} = 0$
while $\hat{D}_\theta(X^{(i)}, Y^{(i)}) > \epsilon$ and $k < \frac{|Y|}{2}$ and $k^{(i)} - k^{(i-1)} > 0$ **do**
  $\theta \leftarrow$ Train$(X^{(i)}, Y^{(i)})$ ▷ Train discriminator
  $\hat{D} \leftarrow \hat{D}_\theta(X^{(i)}, Y^{(i)})$ ▷ Estimate divergence
  $\beta \leftarrow F_\theta(X^{(i)}, Y^{(i)})$ ▷ Estimate feature importance
  $C_i \leftarrow \psi_\tau(\beta, \hat{D})$ ▷ Select corrupted features
  $X^{(i+1)}, Y^{(i+1)} \leftarrow X^{(i)}_{\overline{C_i}}, Y^{(i)}_{\overline{C_i}}$ ▷ Remove detected features
  $k^{(i+1)} \leftarrow k^{(i)} + |C_i|$ ▷ Update detected feature counter
  $i \leftarrow i + 1$
end while
$C = \bigcup_{j=0}^{i-1} C_j$ ▷ Combine all detected features
$C \leftarrow$ Refine(C) ▷ Use knee-locator to refine detected features
return C

DF-Locate (Section 4, Figure 1, and Algorithm 1) is the proposed method within DataFix that localizes the features originating the distribution shift by performing feature selection in an iterative way. First, starting with $i = 0$, and a reference $X^{(0)} = X$ and query $Y^{(0)} = Y$ datasets, a set of discriminators are trained $\theta = \text{argmax}_\theta \hat{D}_\theta(X^{(i)}, Y^{(i)})$. The discriminators are used to predict the empirical total variation divergence (TVD) between distributions $\hat{D} = \hat{D}_\theta(X^{(i)}, Y^{(i)})$, and a feature importance score for each feature $\beta = F_\theta(X^{(i)}, Y^{(i)})$. The divergence and feature importances are used to select potentially corrupted features with the feature removal policy function (see section 4) $C_i = \psi_\tau(\beta, \hat{D})$. Then, the detected features are removed from X and Y, such that $X^{(i+1)} = X^{(i)}_{\overline{C_i}}$ and $Y^{(i+1)} = Y^{(i)}_{\overline{C_i}}$. The process is repeated as long as the estimated divergence is smaller than a threshold $\hat{D}_\theta(X^{(i)}, Y^{(i)}) > \epsilon$, less than half of the features of the dataset are removed $k < \frac{|Y|}{2}$, or at least one feature is selected by the feature removal policy function at each step $k^{(i)} - k^{(i-1)} > 0$. In this work we use $\epsilon = 0.02$ as the stopping threshold. After the iterative process is stopped, a refinement step, described below, is applied to remove features that might have been incorrectly selected as corrupted.

In order to perform the refinement step, we store intermediate steps so that we can revisit and select the optimal stopping point throughout the iterative process. At each iteration, we store the indexes of the features selected as corrupted, the corresponding estimated TVD, and the number of removed features. While the ideal stopping iteration would be determined by the iteration providing the best localization performance, in terms of, for example, F-1 score, this requires access to the ground truth, which is unavailable in practice. However, there often exists a point at which the cost of removing additional features (i.e., selecting non-corrupted features as corrupted) outweighs the corresponding decrease in TVD. Once such optimal iteration is determined, all identified corrupted features up to that iteration are flagged as corrupted and later changed by DF-Correct.

To determine the optimal iteration from the curve depicting the TVD as a function of the total number of removed features, we locate the elbow or knee from a processed version of the curve (Figure 7). A curve obtained from a perfect discriminator would always exhibit a convex and decreasing shape. However, due to the intrinsic randomness in the training and evaluation process of the discriminator,

this ideal shape is not always achieved, making the task of locating the knee challenging. Several techniques can be applied to smooth the curve and transform it into a convex and decreasing function. In our approach, we make use of the Savitzky-Golay filter [72] due to its ability to effectively remove signal noise without distorting the underlying trend of the curve. Additionally, we apply an opening operation to eliminate any local maxima in the curve, ensuring that each point is equal to or smaller than its left neighbor. Furthermore, we process the initial iterations of the curve to enforce strict decreasing behavior.

The Savitzky-Golay filter relies on two key parameters: the length of the filter window and the polyorder used for fitting the samples. For the window length, we define it as $\max(5, 2\lfloor\zeta\delta/2\rfloor + 1)$, where ζ is chosen from the set $\{1, 2, 3, 5, 7\}$, and δ represents the average number of removed features at each iteration. We also experiment with different polyorders, considering values from the set $\{3, 4\}$. Based on our experimentation on the simulated datasets, we determine that the optimal choice for ζ is 2, and the best polyorder is 4. To identify the knee point, we employ the knee locator method introduced by [73]. The knee locator involves two primary parameters: the sensitivity (S) and the online parameter. The sensitivity parameter determines the number of "flat" points we anticipate encountering in the unmodified data curve before identifying a knee, while the online parameter enables the correction of previous knee values when set to $True$. We explore different values for S from the set $\{1, 3, 5, 7\}$, and for the $online$ parameter from the options $\{True, False\}$. Our experimentation reveals that the optimal values for S and $online$ are 5 and $False$, respectively.

Figure 7 shows an example of the knee location, used to refine the selection of features. The F-1 score can serve as a way to evaluate the selected stopping point, showing the trade-off between reduced TVD and the number of removed features.

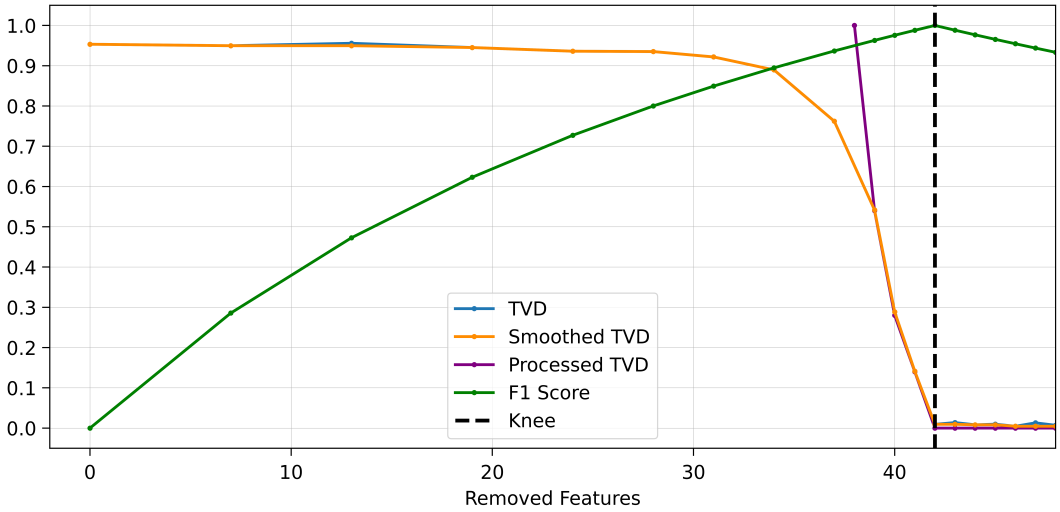

Figure 7: Knee location based on distribution shift: TVD vs. removed Features and F-1 score in location task.

E DF-Correct

DF-Correct (Section 5, Figure 2, and Algorithm 2) is the proposed method within DataFix that corrects the feature shifts by changing the values of the features in Y originating the distribution shift through an iterative approach. Given the set of corrupted features C, DF-Correct tries to generate a new query dataset Y', such that $Y_{\overline{C}} = Y'_{\overline{C}}$, while $D(X, Y') < D(X, Y)$. DF-Correct starts by obtaining an initial candidate of Y' by setting the values within the subset C of Y as missing and performing imputation with linear regression and k-NN. Furthermore, a naive initial candidate is generated by replacing the corrupted values of Y with randomly selected values of X (restricted to the features subset C). Note that this naive initialization already fixes distribution shifts where features are completely independent of each other. The set of the 3 initial candidates, $V = \{Y^0, Y^1, Y^2\}$, is evaluated by computing the empirical total variation divergence with a classifier, and the one providing the lowest empirical divergence is selected as the initial corrected

Algorithm 2 DF-Correct: Feature Shift Correction

Inputs:
  X; ▷ Reference
  Y; ▷ Query
  C; ▷ Corrupted Features
  ϵ; ▷ Divergence Threshold
$V = \{Y^0, Y^1, Y^2\} \leftarrow$ Impute(X, Y, C)
$Y' \leftarrow \text{argmin}_{Y^i \in V} \hat{D}_\theta(X, Y^i)$
if $\hat{D}_\theta(X, Y') < \epsilon$ **then**
  return Y'
end if
for epoch **do**
  $\theta \leftarrow$ Train(X, Y') ▷ Train discriminators
  $I \leftarrow$ DetectIncorrect$(\mathcal{D}_\theta(Y'))$ ▷ Detect samples that require feature correction
  $B \leftarrow$ GenerateProposals(X, Y', V, C) ▷ Obtain proposals from X, Y', V
  for $i \in I$ **do**
    $b_i \leftarrow \text{argmax}_{b \in B}\, r_{\theta_i}(y_i^{(b)})$ ▷ Find best proposal
    $Y' \leftarrow$ update$(Y', y_i^{(b_i)})$ ▷ Update dataset with the corrected sample $y_i^{(b_i)}$
  end for
  if $\hat{D}_\theta(X, Y') < \epsilon$ **then**
    return Y'
  end if
end for
return Y'

query $Y' = \text{argmin}_{Y^i \in V} \hat{D}_\theta(X, Y^i)$. If $\hat{D}_\theta(X, Y') < \epsilon$, the correction process is done and Y' is returned. Otherwise, if $\hat{D}_\theta(X, Y') > \epsilon$, an iterative process where some samples of Y' are modified is performed.

The iterative process tries to find new values of $Y' = \{y'_1, y'_2, ..., y'_{N_y}\}$ that reduce the empirical total variation distance:

$$\hat{D}_\theta^{TV}(X, Y') = \frac{1}{N_x} \sum_{i=1}^{N_x} g(r_\theta(x_i)) - \frac{1}{N_y} \sum_{j=1}^{N_y} g(r_\theta(y'_j)) \tag{13}$$

with $g(u) = \frac{1}{2}\text{sign}(u - 1)$. Because the values of X are left untouched, this becomes equivalent to solving:

$$Y' = \underset{Y}{\text{argmin}} \max_\theta \sum_{j=1}^{N_y} -g(r_\theta(y'_j)) = \underset{Y}{\text{argmax}} \max_\theta \sum_{j=1}^{N_y} g(r_\theta(y'_j)) \tag{14}$$

In order to perform such an optimization process, a set of classifiers are trained using X, Y', and data augmentation consisting of performing random permutations within the features of the reference X dataset, in order to generate extra samples of "corrupted" sequences. After training the classifiers, the next step is to find which samples need to be corrected. Note that when $p = q$, we have $\mathbb{E}[\hat{D}_\theta^{TV}(X, Y)] = 0$, and:

$$\mathbb{E}[\sum_{j \in N_y} g(r_\theta(y_j))] = \mathbb{E}[\sum_{m \in N^+} g(r_\theta(y_m))] - \mathbb{E}[\sum_{n \in N^-} g(r_\theta(y_n))] = 0 \tag{15}$$

where $N^+ = \{i : r_\theta(y_i) > 1\}$ are the indices of samples classified as positive by the discriminator, and $N^- = \{i : r_\theta(y_i) < 1\}$ are the indices of samples classified as negative. Furthermore, both sets have, in expectation, the same size $\mathbb{E}[|N^+|] = \mathbb{E}[|N^-|] = \mathbb{E}[\frac{|N_y|}{2}]$. This indicates that when both distributions are equal, a discriminator will approximately classify half of the samples as positive and

half as negative. Therefore we only correct the set of samples I, including up to $\frac{|N_y|}{2}$ samples with the highest probability of being corrupted:

$$I = \{i : r_{\theta_i}(y_i) < r_{\theta_{i+1}}(y_{i+1}) < 1\} \tag{16}$$

with $|I| \leq \frac{|N_y|}{2}$. Then, we construct a set of feature value proposals B that will replace the corrupted features. This proposal set is constructed by including within B all the feature values of the reference X, of the initial imputed candidates V, of the current corrected query Y', and random permutations of X. Then, for every sample $i \in I$, each of the proposals $b \in B$ is placed as an alternative to the corrupted features, generating a sample $y_i^{(b)}$, where ${y_i^{(b)}}_C = b$, and ${y_i^{(b)}}_{\overline{C}} = y_{i\overline{C}}$. The proposal providing the highest probability of being "non-corrupted" is selected:

$$b_i = \underset{b \in B}{\operatorname{argmax}}\, r_{\theta_i}(y_i^{(b)}) \tag{17}$$

Finally, the updated sample $y_i^{(b)}$ is placed inside the corrected query Y'. After updating all samples in I, the divergence is computed again, and if $\hat{D}_\theta(X, Y') > \epsilon$, the process is repeated for a number of epochs. Typically, the number of epochs is set to 1 or 2, as the correction process can become computationally intensive for large datasets (see next sections).

F Computational Time

Large and high-dimensional datasets are becoming the norm, therefore, methods that detect and correct feature shifts should be able to properly scale with respect to the number of samples and features.

Figure 8 presents the average computational time of feature shift location benchmarking methods as a function of the product between the number of samples and features for each dataset. MI and selectKbest stand out as the fastest benchmarks (using the Chi-square test for categorical datasets and ANOVA-F test for continuous datasets). MRMR and FAST-CMIM, although performing adequately in terms of speed for small datasets, encounter challenges in scaling with larger dataset sizes. Consequently, they fail to produce results within the 30-hour time limit for Founders and Canine datasets. Furthermore, the feature-shift detection techniques KNN-KS and, particularly MB-KS, demonstrate significantly slower performance, rendering them incapable of delivering results within the given time constraint for Founders, Canine, Dilbert, and Phenotypes datasets. DF-Locate proves to be a reasonably efficient method, exhibiting good scalability as the dataset size increases, while providing the best localization performance.

In Figure 9, we conduct a comparative analysis of the computational time for shift correction methods. Although DF-Correct is not faster than simpler techniques like median or linear regression, its speed surpasses several competing methods, such as MIRACLE and HyperImpute. Furthermore, DF-Correct exhibits reasonable runtime even for the largest high-dimensional Canine dataset, successfully correcting the distribution shift within the 30-hour time limit. Many of the competing methods were unable to provide results for the Canine dataset due to their excessive time complexity and/or memory requirements. Therefore, DF-Correct provides the best correction in terms of distribution shifts, while still providing competitive or faster speeds than competing methods.

G Extended Feature Shift Localization Results

We evaluate the performance of DF-Locate across different fractions of manipulated features: 5%, 10%, and 25%. The F-1 scores, depicted in Figure 10, were computed by averaging across manipulation types and taking the median across real datasets for each feature shift localization method. Our findings show that DF-Locate obtains consistently higher performance, irrespective of the fraction of manipulated features involved. However, it is important to note that this is not the case for other methods, such as MRMR and FAST-CMIM, which exhibit limitations in their location capabilities, particularly when there are only a few corrupted features. Note that when a small percentage of manipulated features is present, a smaller amount of features need to be localized as corrupted. However,

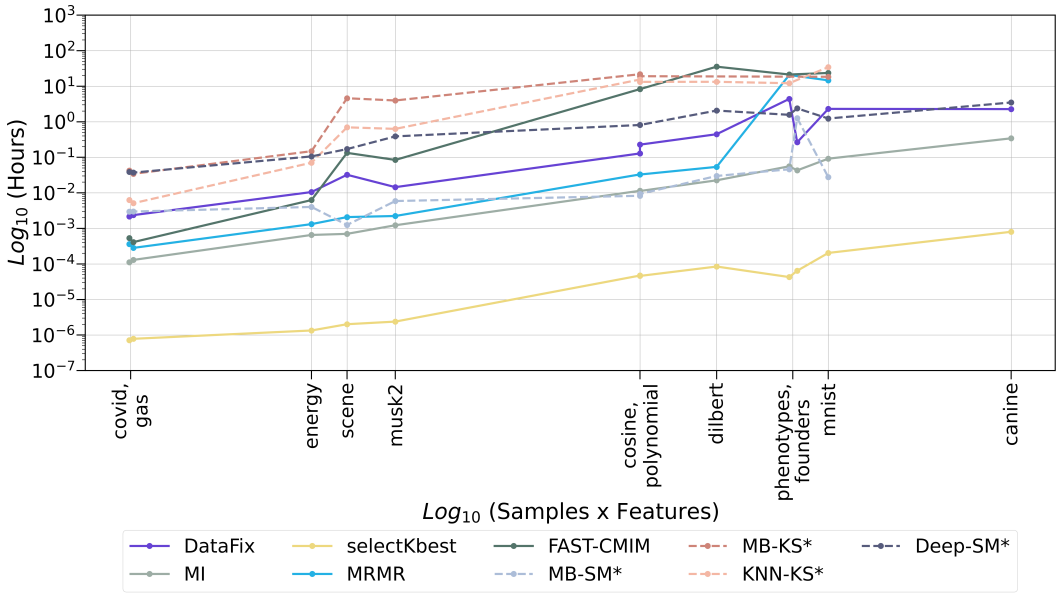

Figure 8: Computational time for shift location methods based on real dataset size.

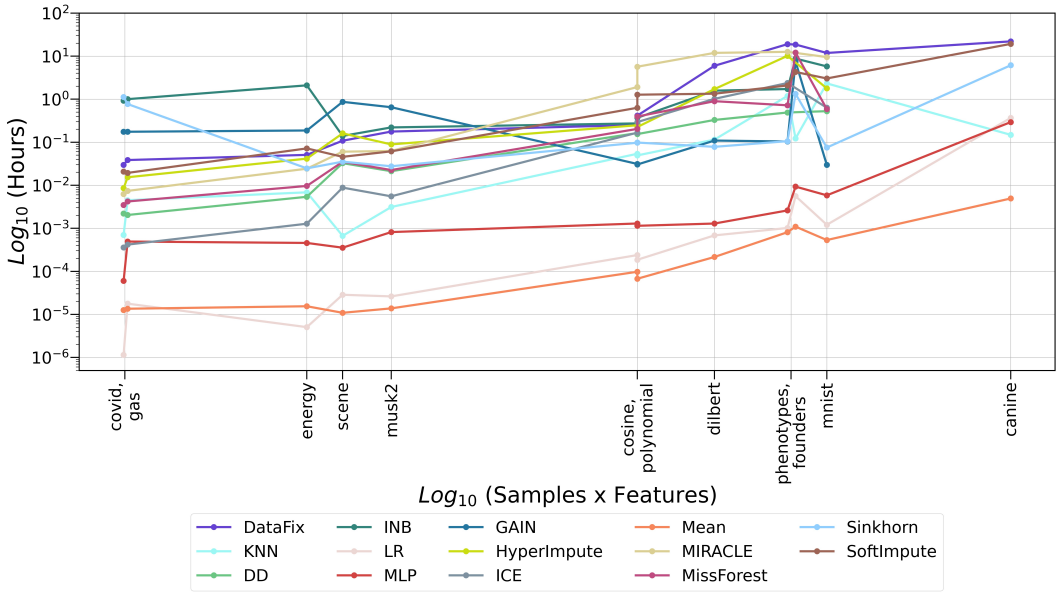

Figure 9: Computational time for shift correction methods based on real dataset size.

this could also lead, in some cases, to lower distribution shifts, making the detection of such shifts more challenging. On the other hand, the presence of a larger percentage of manipulated features can make the localization task more challenging, while the empirical detection of the presence of the shift can be, in some cases, easier.

Figure 11 shows the mean F-1 score of DF-Locate and various shift location methods applied to the real datasets. The symbol 'x' denotes experiments that are missing due to exceeding the time limit of 30 hours. Note that most methods fail to process high-dimensional datasets such as Founders and Canine, while DF-Locate is able to provide accurate results while scaling to large datasets. DF-Locate consistently exhibits superior performance compared to all benchmarking methods across most datasets. The only exception observed is with the Phenotypes dataset, where selectKBest slightly outperforms DF-Locate in locating the corrupted features. Nevertheless, for the remaining datasets, DF-Locate surpasses all competing approaches and is only slightly surpassed or equaled

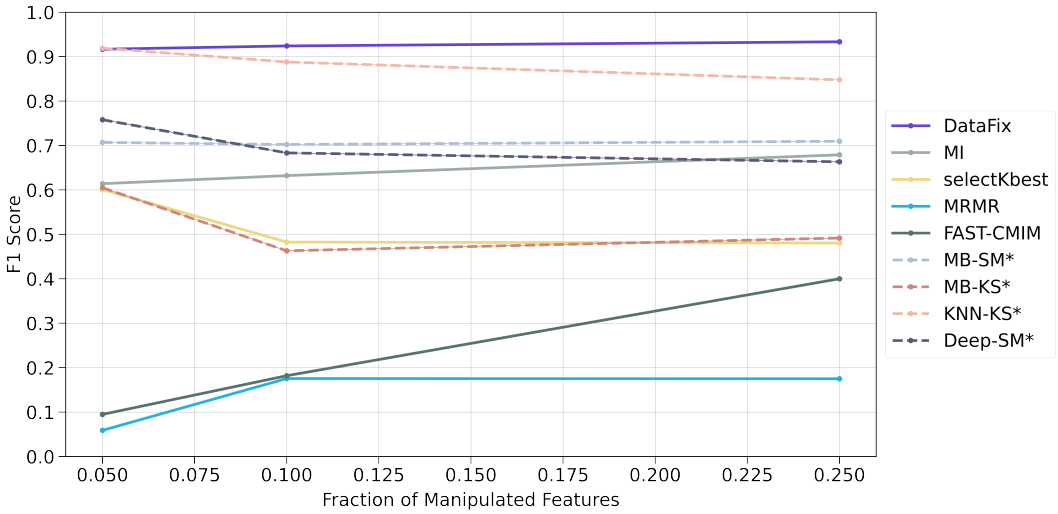

Figure 10: Median F-1 scores of shift location methods by fraction of manipulated features on real datasets.

in performance on a few occasions by Deep-SM (*) or KNN-KS (*), both of which use the ground truth $|C|$. Additionally, it is worth noting that MI always outperforms selectKbest on datasets with continuous features, while the opposite holds true for datasets with categorical features.

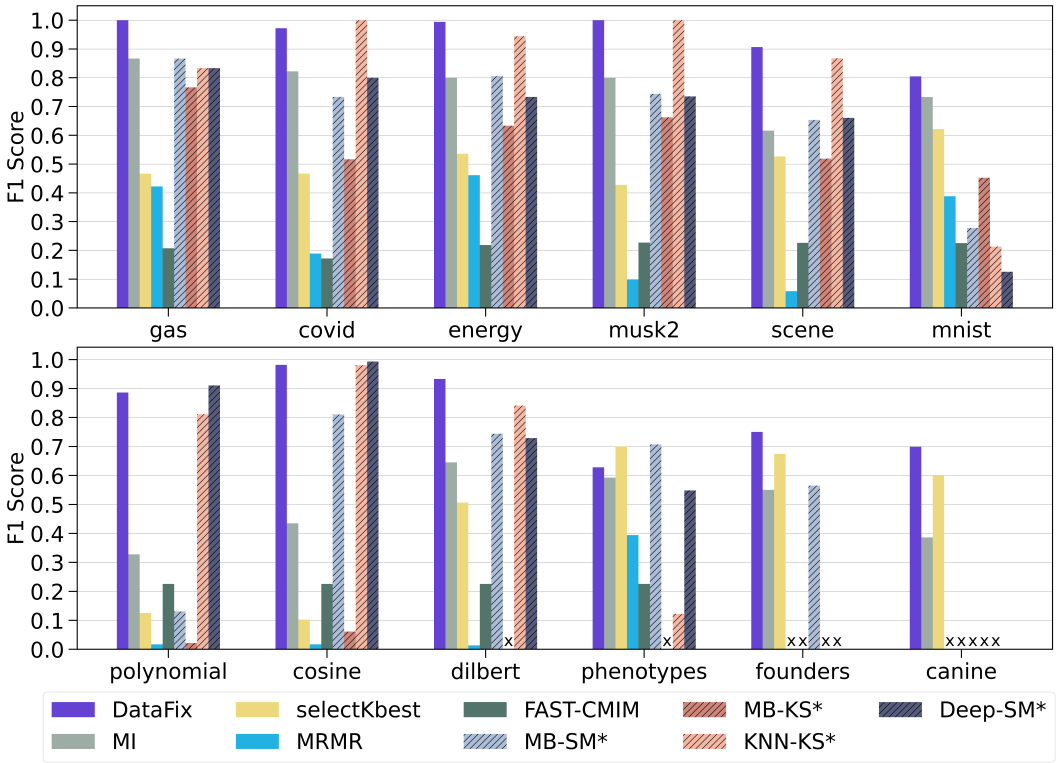

Figure 11: Mean F-1 scores of shift location methods by real datasets. 'x' indicates missing experiment. Higher is better.

Figure 12 shows the median F-1 scores of DF-Locate and other competing methods, categorized by the feature manipulation type applied to the real datasets. The average F-1 score is computed

across fractions of manipulated features, followed by the computation of the median F-1 score across different datasets. The symbol 'x' is used to indicate missing experiments for MB-KS and manipulation types 6.1-6.3 and 10. These manipulations are applied to categorical datasets only (Phenotypes, Founders, Canine), and the MB-KS method did not yield results for any of these three datasets within the specified time constraint of 30 hours. Manipulations involving shifts caused by the correlation between features (manipulations 3 and 8) are not detected by methods such as MI, selectKbest, MRMR and Fast-CMIM, while being accurately detected by MB-SM, KNN-KS, Deep-SM, and our proposed method. Note that while manipulation type 8 breaks the theoretical equivalence between feature selection and feature shift localization problem (see previous sections), it is still accurately localized by DataFix. Manipulation 10, consisting of replacing feature values with the ones predicted with a k-NN, is the most challenging to detect by DataFix. Note that such manipulation can lead, in some scenarios, to small or undetectable distribution shifts, as k-NN provides perfect predictions when its training dataset size goes to infinity.

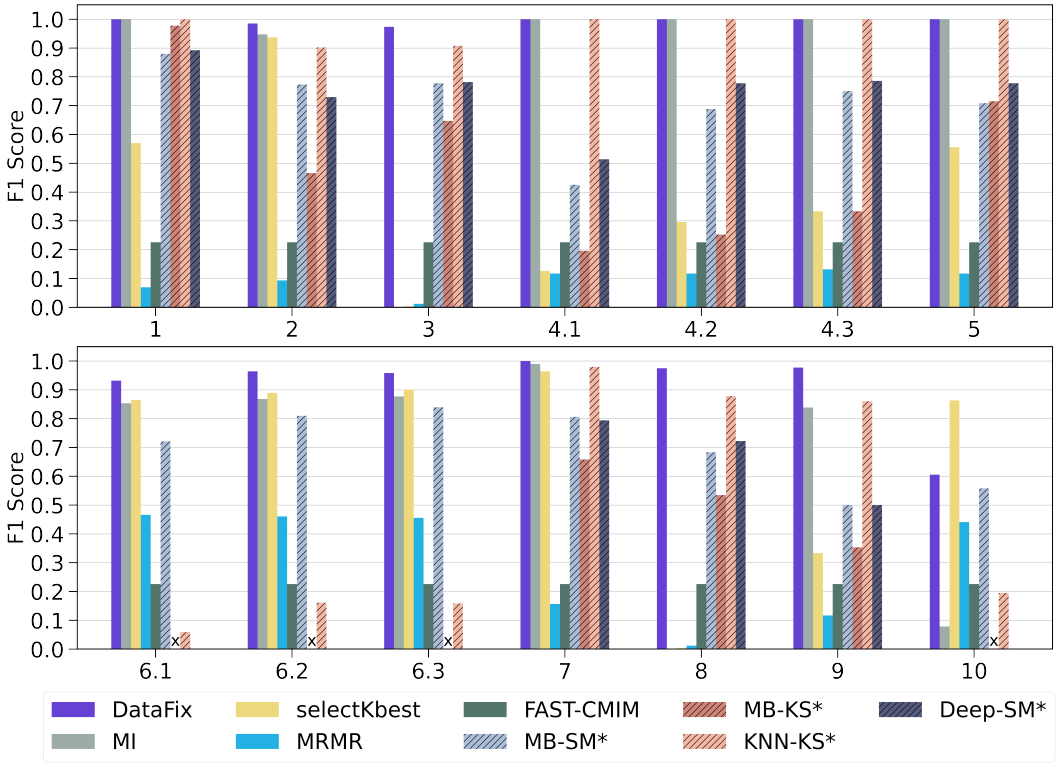

Figure 12: Median F-1 scores of shift location methods by feature manipulation type on real datasets. 'x' indicates missing experiment. Higher is better.

Figure 13 displays the mean F-1 scores of shift location methods by simulated datasets. Similar to the real datasets, DF-Locate outperforms or matches all competing methods, except in datasets 7 and 8, where MB-SM and Deep-SM obtain a higher F-1 localization score when using $|C|$ as extra ground truth information. Note that in a fair comparison where $|C|$ is not used, MB-SM and Deep-SM perform poorly (see Figure 3). DF-Locate obtains lower F-1 scores in simulated datasets 5, 7, and 8, with respect to the other datasets, which involve shifts originated by a mismatching correlation between distributions.

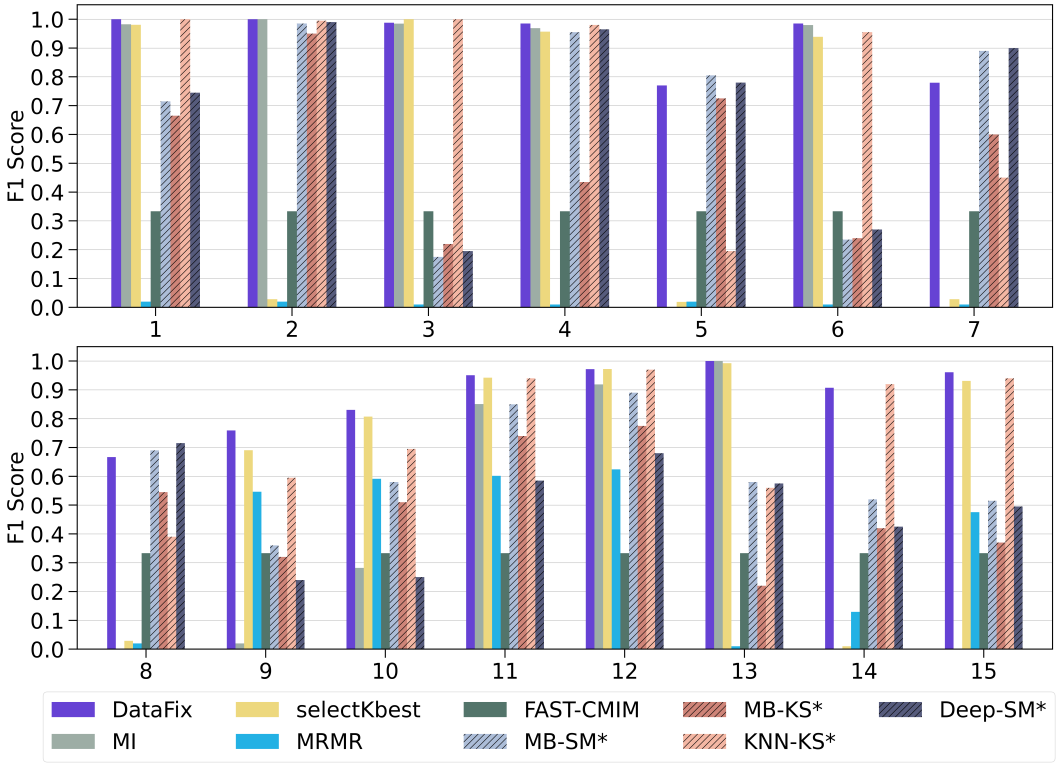

Figure 13: Mean F-1 scores of shift location methods by simulated datasets. Higher is better.

H Extended Feature Shift Correction Results

Figures 14 and 15 provide a comprehensive evaluation of the performance of DF-Correct and competing shift correction methods across real datasets, by using the evaluation metrics W_2^2, D_{hp}, and D_{skl}. The reported metrics provide empirical estimates of the divergences between the corrected query datasets and the reference datasets. Note that first, the empirical divergences between the reference and query dataset (prior to any manipulation) are computed, and subtracted from the reported metrics. In terms of empirical divergences, DataFix outperforms all other methods by a significant margin for most datasets, demonstrating its high effectiveness in providing corrected query datasets that are close to the reference distribution. Following DataFix, simpler methods like k-NN and linear regression demonstrate competitive performance for most datasets, while benchmarks such as ICE, HyperImpute, INB, Sinkhorn, and MLPs yield favorable results for specific datasets. Furthermore, DataFix consistently achieves the lowest W_2^2 values across most datasets, with exceptions being MNIST and Phenotypes. Specifically, for the MNIST dataset, k-NN, MLP, HyperImpute, and ICE surpass DataFix achieving the best W_2^2 value of 0. Note that for high-dimensional datasets such as Dilbert, Phenotypes, Founders, and Canine, the W_2^2 metric saturates to 0 for multiple methods, making the other metrics a better alternative to compare the quality between methods. DataFix is outperformed by MIRACLE in terms of D_{hp} and D_{skl} in the Dilbert and Phenotypes dataset, however, it is important to note that Miracle's slow execution time prevents it from providing results for the Founders and Canine datasets.

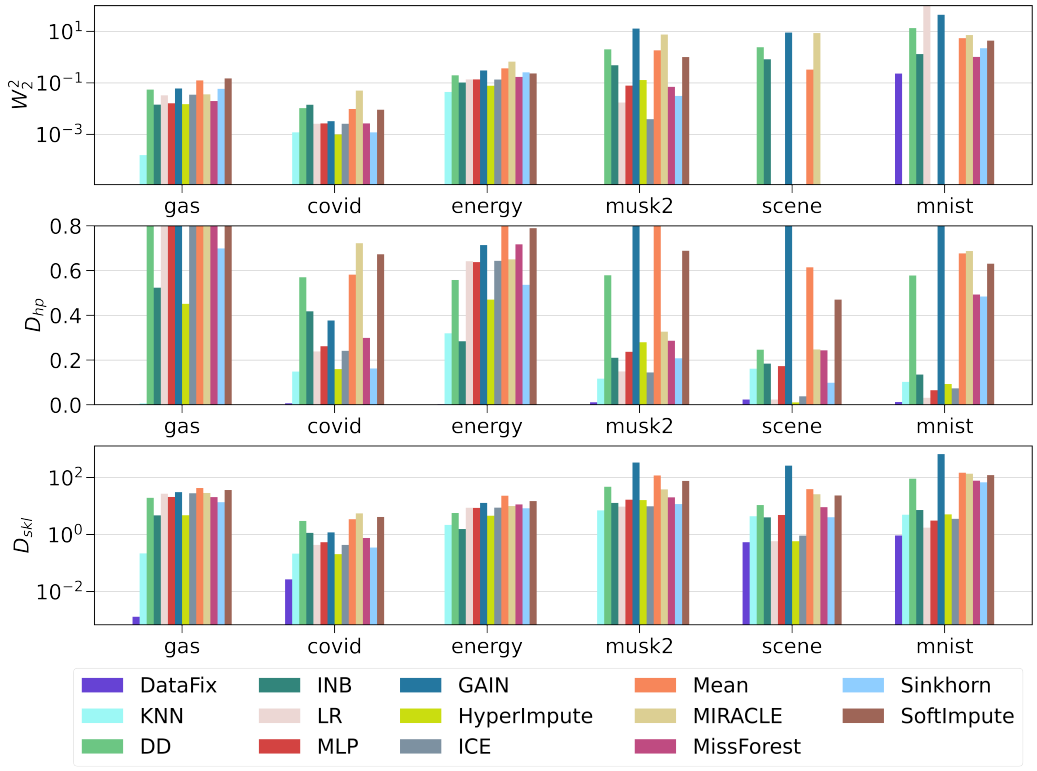

Figure 14: W_2^2, D_{hp}, and D_{skl} of shift correction methods by real datasets. Lower is better. (Part 1)

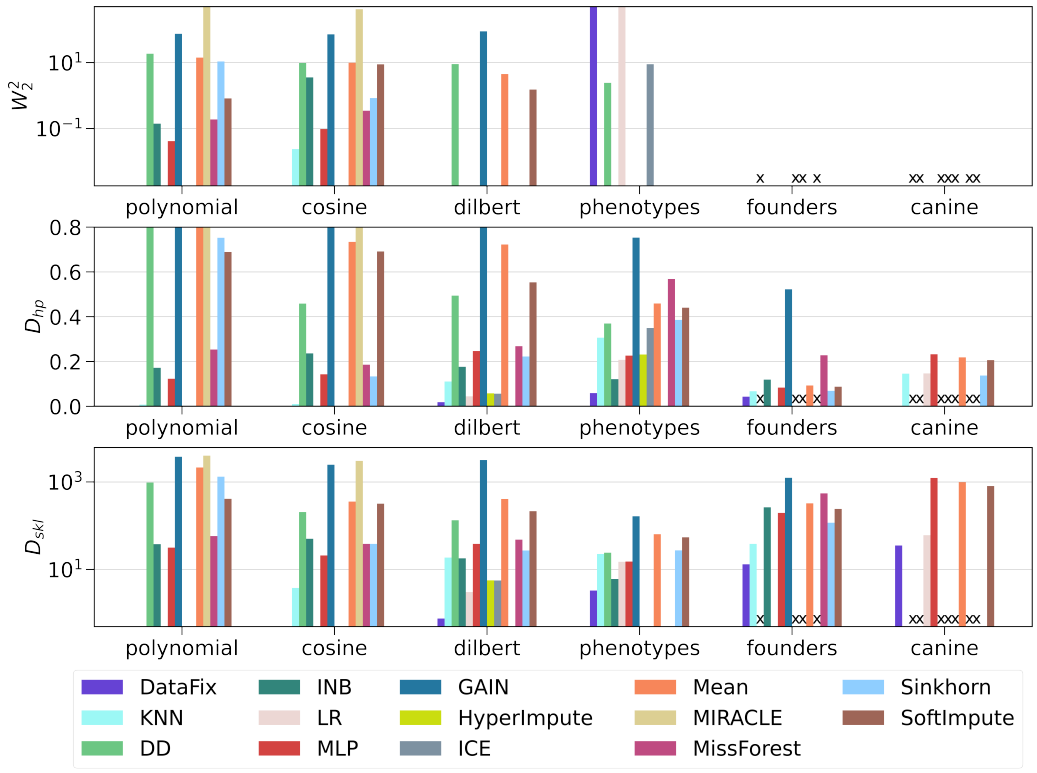

Figure 15: W_2^2, D_{hp}, and D_{skl} of shift correction methods by real datasets. Lower is better. (Part 2)

I Classifier Analysis for Localization and Correction

Figure 16 provides the F-1 score results for DF-Locate when using different classifiers as discriminators within the iterative process applied to the simulated datasets. Tree-based methods including Random Forest (RF), CatBoost, ExtraTree, and LightGBM (LGBM) provide highly similar results, with high F-1 scores, surpassing linear models such as logistic regression (LogReg) and a support vector classifier (SVC). We selected RF as our discriminator as it provided much faster training times while being highly competitive in localization accuracy.

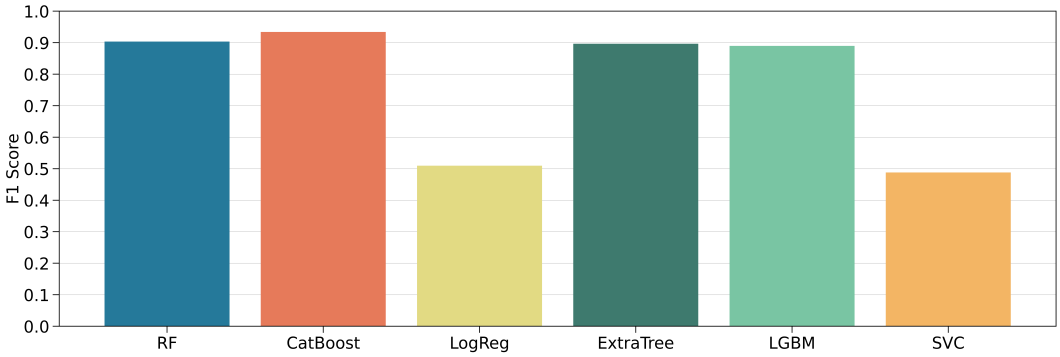

Figure 16: Mean F-1 scores by DF-Locate using different classifiers on simulated datasets. Higher is better.

Figure 17 provides the feature shift correction metrics for DF-Correct when using different classifiers as discriminators within the iterative process within the real and simulated datasets. Similar to DF-Locate, Tree-based methods provide similar results, surpassing the linear methods in most metrics. We use D_{hp} as a metric to select our method because it provides an estimate that tightly bounds the total variation distance. Namely, we select CatBoost as our discriminator as it provides competitive performance with the other tree-based methods in the simulated datasets, and clearly outperforms the others in the real datasets.

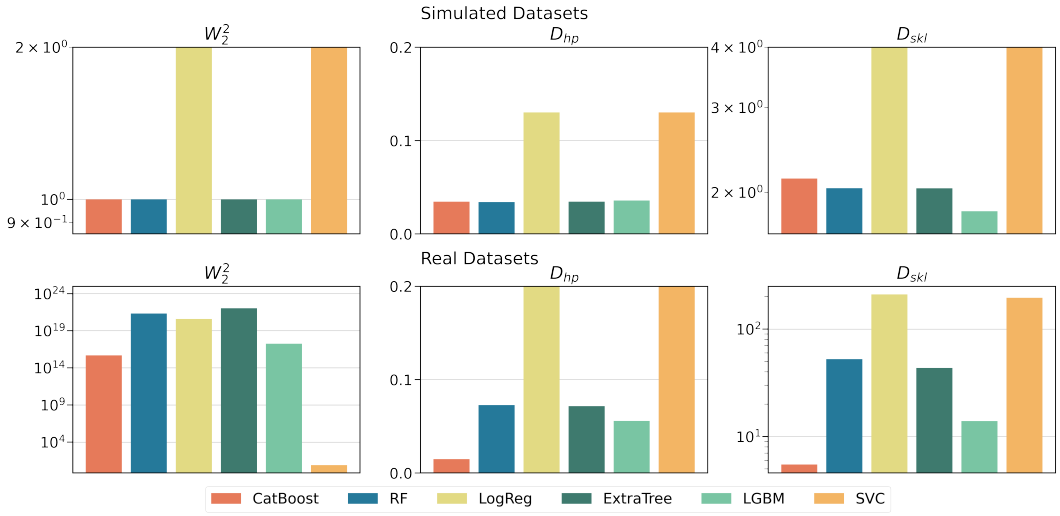

Figure 17: Mean W_2^2, D_{hp}, and D_{skl} by DF-Correct using different classifiers on simulated (top) and real (bottom) datasets. Lower is better.

J DataFix Analysis

Figures 18 and 19 provide two additional examples that illustrate the iterative process of DF-Locate before and after shift correction in simulated datasets 1 and 2, respectively. In each dataset, there are 200 corrupted features out of 1000. Similar to Figure 5, Figures 18 and 19 (left) display the TVD estimated by the random forest (blue), which provides a lower bound for its ground truth Monte Carlo estimate (black), as the iterative process detects and removes corrupted features. The F-1 detection score progressively increases until all corrupted features are identified, leading to the termination of the iterative process. Figures 18 and 19 (right) showcase the iterative process applied to the corrected query using different methods.

In simulated dataset 1 (Figure 18), both MIRACLE and MLP yield an updated query that results in a lower empirical divergence. However, for simulated dataset 2 (Figure 19), only MLP produces an updated query that leads to a lower empirical divergence. The other shift correction benchmarking methods generate an updated query that increases the shift instead of reducing it. Notably, DF-Correct (Purple) provides a precisely corrected query with no empirical divergence detected by DF-Locate.

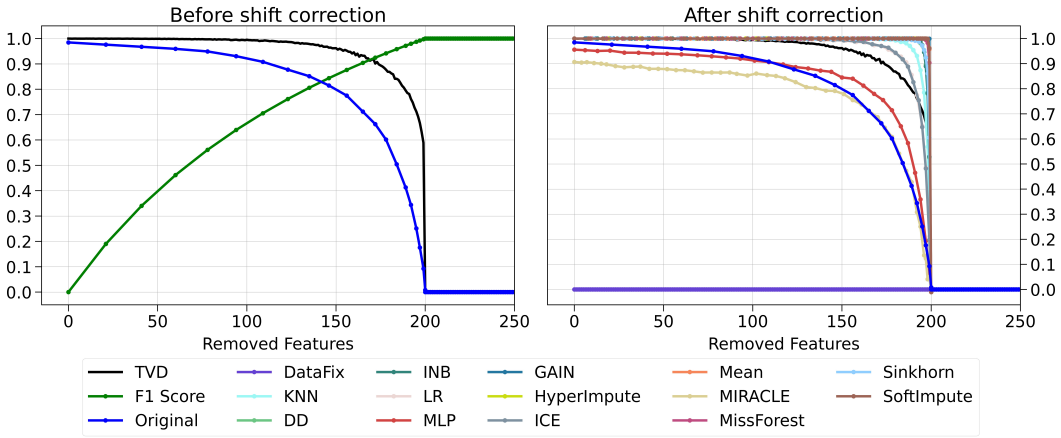

Figure 18: DF-Locate iterative process before and after shift correction.

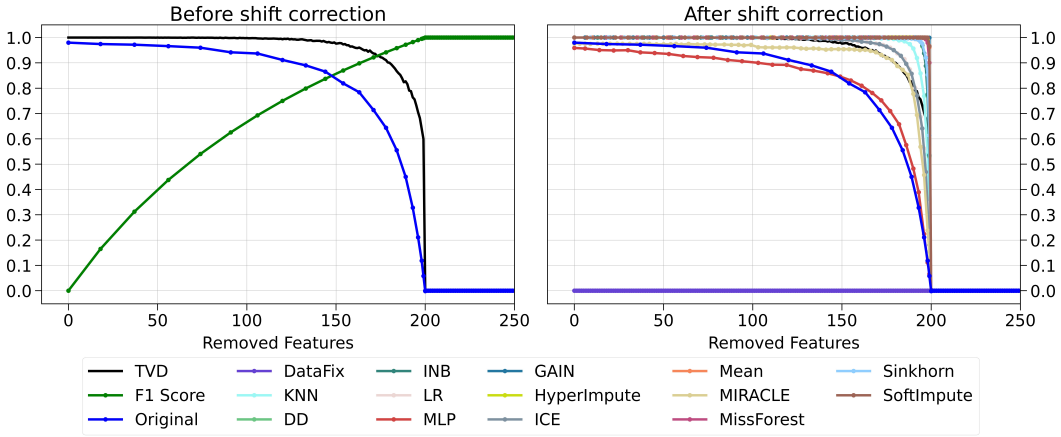

Figure 19: DF-Locate iterative process before and after shift correction.