ML Works - Drift Documentation

1. Introduction to Drift:

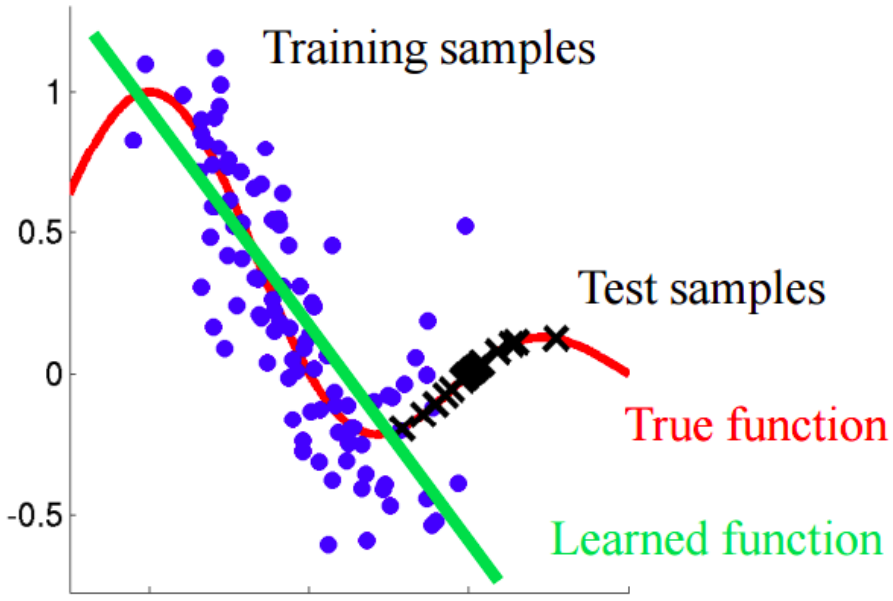
In a typical setting when a machine learning model is developed to solve a business problem, the ML model is fit on the training data set. It is then deployed to run inference on production data. Model performance in production degrades over time. This happens due to multiple reasons such as:

- Production data has a different data distribution compared to the training dataset. In other words, the training data chosen is not representative of the entire population
- Relationship between features (independent variable) & target (dependent variable) changed over time.

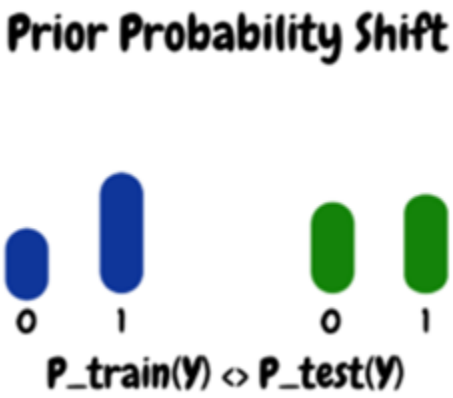
2. Types of Drift

Drift can be broadly classified into the following categories:

- **2.1 Covariate Shift -** To understand using an example, consider a churn prediction model where the training data has people above 25 years of age but in production, we also see people below 25 years of age. This could cause the model to behave unexpectedly if the people below 25 are more likely to churn in favor of cheaper offers elsewhere.
    - Formal definition: When only the distributions of covariates **X** change and everything else is the same. Formally it is defined only for $X \rightarrow Y$ problems as $P_{train}(y|x) = P_{test}(y|x)$ but $P_{train}(x) != P_{test}(x)$.



- **2.2 Prior Probability Shift -** Indicates that the distribution of the target variable - **Y** changed in production.
    - Considering a spam detection model, if in training data we have an even distribution of spam and not spam, i.e. 50% each but in production, the distribution changes and becomes 80% spam and 20% not spam.
    - Formal definition: Prior probability only appears in Y → X problems or Generative Machine Learning algorithms where we learn the joint probability distribution P(x,y). Formally it is defined as the case where $P_{train}(x|y) = P_{test}(x|y)$ but $P_{train}(y) != P_{test}(y)$.



- **2.3 Concept Drift -** Predicting housing prices before and after the COVID-19 pandemic would suffer from concept drift as the relationship is expected to change.
    - Formal definition: Concept drift occurs when the relationship between the input and the target variable has changed. Formally it can be defined as the case when:
        - $P_{train}(y|x) != P_{test}(y|x)$ but $P_{train}(x) = P_{test}(x)$ in X->Y problems
        - $P_{train}(x|y) != P_{test}(x|y)$ but $P_{train}(y) = P_{test}(y)$ in Y->X problems

3. Measuring Drift

ML Works community version supports drift detection for structured data only and **measures covariate shift.**

Domain Classifier

Considering training & testing/production as two different domains, a binary classifier is trained to identify drift between a source (train) and target (test) dataset.

If the classifier can easily distinguish whether each observation belongs to source or target then there is a very high drift between the two datasets; but, if the classifier is not able to distinguish between the two then we can say that two distributions are similar and there is very low drift between the two datasets.

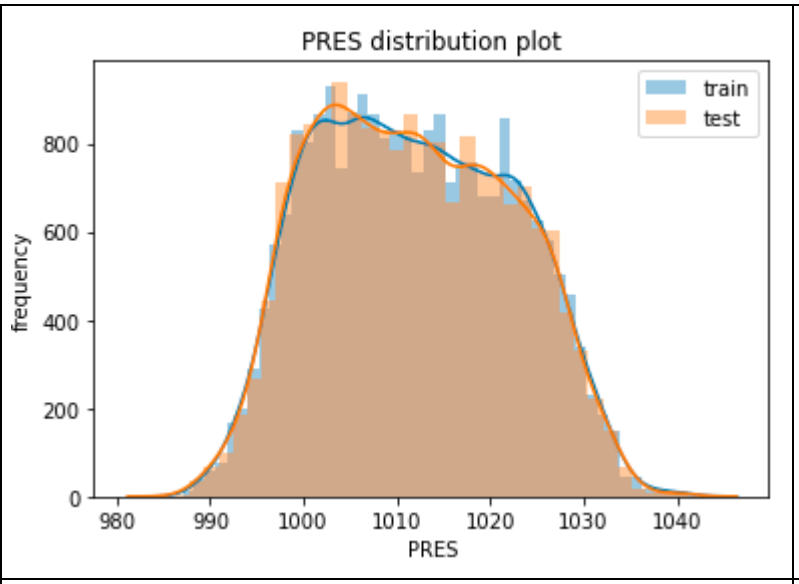
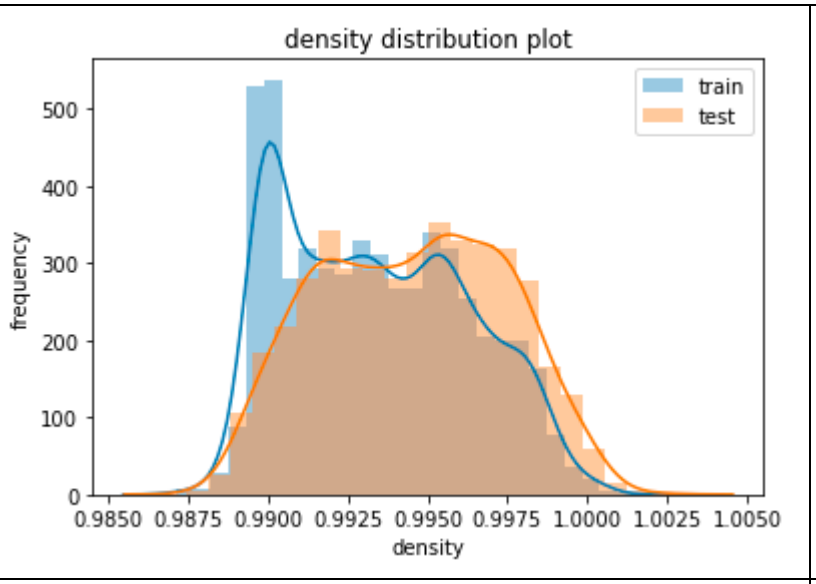
Our Implementation

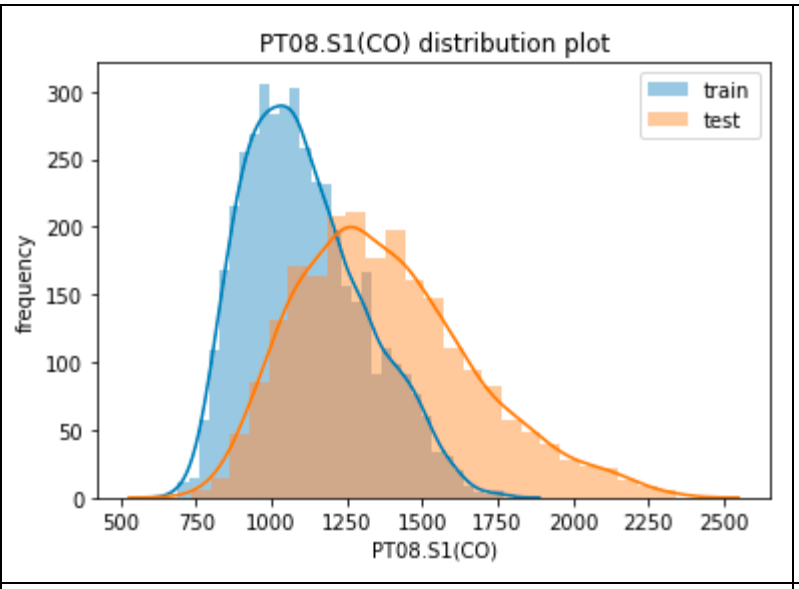
ML Works drift library provides drift values at both overall level and feature level:

- **Overall Drift** - Overall drift represents how the entire dataset (all features) has collectively drifted. It is calculated by taking into consideration all the features in the dataset. The performance of the classifier is measured using the AUC_ROC score which is further translated to the drift score.
- **Feature Drift** - The same idea of overall drift can be applied to each feature in the dataset to calculate feature level drift score.

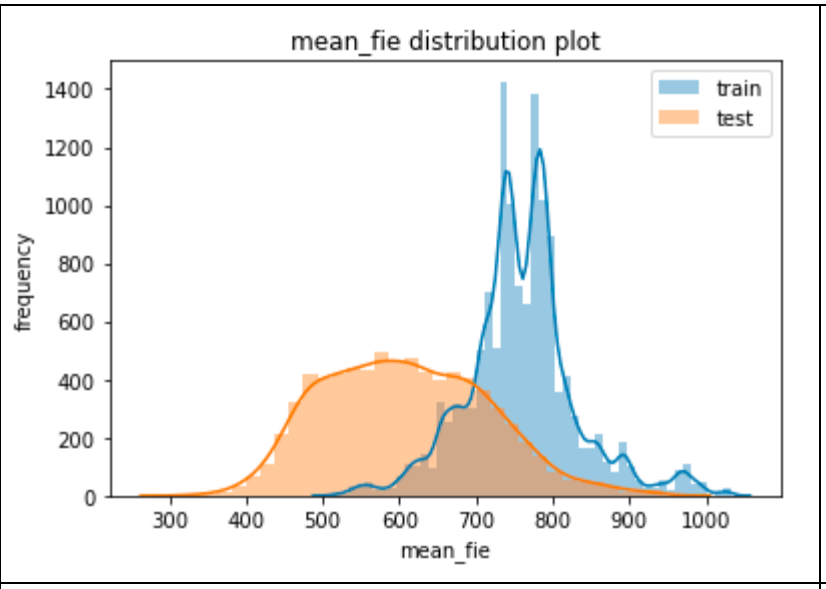
4. Drift & data distribution visualization

Data distributions can be visualized for intuitive estimations of drift (covariate shift drift)

<table>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>PRES - 0% Drift</td><td>Density - 20% Drift</td></tr>
</table>



PT08.S1(CO) - **50% Drift**



Mean Fie - **75% drift**

Feature drift visualization

At the feature level, data distribution can be visualized using histogram plots as shown in the figure below. Each subplot shown is from a different dataset. In each subplot, the X-axis contains the value of a particular feature and the Y-axis contains frequency count. The degree of overlap defines dataset drift (covariate shift). Lesser the overlap, the higher the drift and vice versa. The plots are in the order of increasing drift detected. In the first plot, the PRES feature distribution is identical in both the train & test dataset. Therefore, zero drift. However, in the last plot, GTEP feature, test set distribution is moved right to train set resulting in 75% drift.

For the first two plots, there is a significant overlap between the two distributions hence low drift. The third plot has medium overlap and hence medium drift. The last plot has very little overlap and hence, high drifts.

5. Enterprise version expansion:

Among many of the features offered as part of the Enterprise version of ML Works Drift library, the following are key highlights:

5.1 Feature contribution to Drift

Feature contribution is measure in percentage of how much a feature contributes to overall drift. It helps in recognizing which features are contributing most to the overall drift. The concept of Shapley values (marginal contribution) from game theory is used to calculate feature contribution. The contribution percentages act as weights for evaluating overall weighted drift.

$$contribution_i(in\ \%) = \frac{overall\ drift - \ overall\ drift\ without\ i^{th} feature}{overall\ drift} * 100$$

The function drops one feature from the dataset and calculates the overall drift of the remaining features to calculate the contributions by the remaining feature. The difference in drift value (overall drift with the feature - overall drift without the feature) then yields contribution as a percentage.

5.2 Overall weighted drift:

When the model can easily distinguish the source of a dataset, higher is the drift between the datasets, and vice-versa. The quantitative value of this uneasiness is the overall drift value of the testing dataset w.r.t. training dataset.

$$Overall\ weighted\ drift = \sum_{i=0}^{n} feature\_drift_i * contribution_i$$

$where$:

$feature\_drift_i = drift\ of\ i^{th} feature$

$contribution_i \ = drift\ contribution\ of\ i^{th} feature$

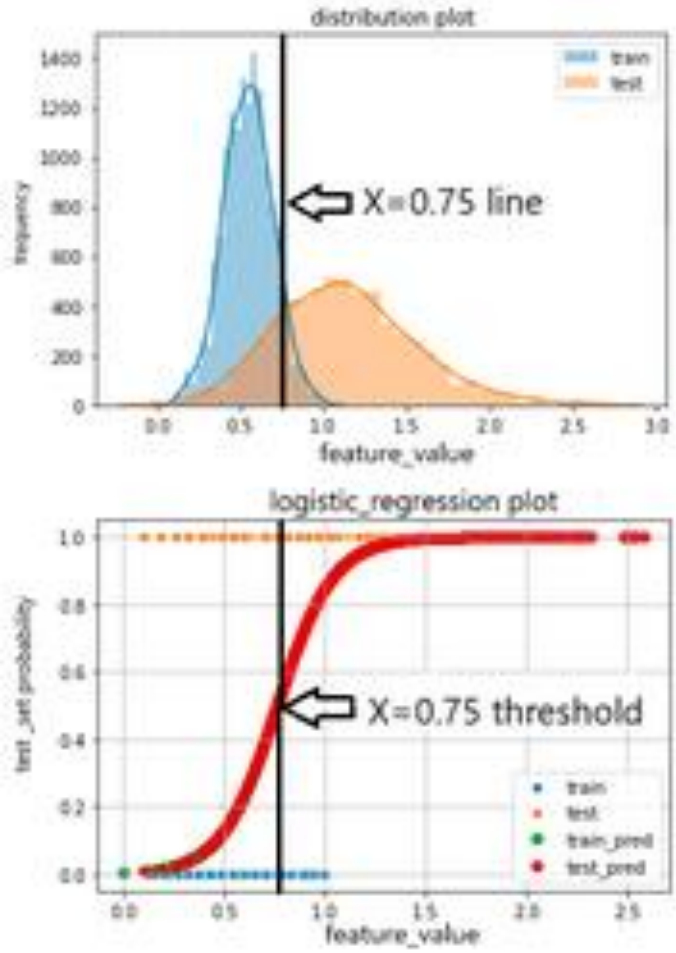
To evaluate overall weighted drift value: For each feature, the product of a feature drift and its corresponding drift contribution, and the sum of this weighted score is the overall weighted drift.

Overall weighted drift is drift which can be explained due to drift in individual features. The remaining drift can be attributed to the overall distribution of data in the dataset, leading to a mismatch between overall drift and overall weighted drift.

5.3 Drift Confidence Score:

The decision boundary of the trained classifier would provide confidence to the developer & consumer. Each of the drift score will be tagged with associated accuracy or confidence metric for more visibility for the end user.

The predicted output for the train & test data-points is plotted where each value is the predicted probability of a point in train set belonging to the test distribution



References:

1. http://iwann.ugr.es/2011/pdf/InvitedTalk-FHerrera-IWANN11.pdf
2. https://rtg.cis.upenn.edu/cis700-2019/papers/dataset-shift/dataset-shift-terminology.pdf