WELLS FARGO

Model Diagnostics Trilogy:

Error and Resilience, Prediction Uncertainty, Bias and Fairness

Aijun Zhang, Ph.D.

Corporate Model Risk, Wells Fargo

Model Validation Class, UNCC Master of Data Science, Fall 2023

Disclaimer: This material represents the views of the presenter and does not necessarily reflect those of Wells Fargo.

Biographical Sketch

- Aijun Zhang is a senior vice president, quantitative analytics senior manager at Wells Fargo. He leads a machine learning & validation engineering team in Corporate Model Risk, responsible for PiML (Python interpretable machine learning) toolbox and VoD (Validation-on-Demand) platform.
- Aijun holds PhD degree in Statistics from University of Michigan at Ann Arbor, and he has 10+ years of experience working in financial risk management.
- Prior to joining Wells Fargo, Aijun was a tenure-track assistant professor at Department of Statistics and Actuarial Science, The University of Hong Kong. He has published ~40 papers in professional conferences and journals, with research topics in interpretable machine learning, data science and statistics.

Machine Learning Model Diagnostics

- **Model performance is not all you need.** ML model performance is often measured by **accuracy,** as examined via standard overall metrics (e.g. MSE, MAE, R2, ACC, AUC, F1-score, Precision and Recall).
- However, model risk assessment by single-valued metrics is insufficient. More granular diagnostics and outcome testing are needed:
  - **Resilience test**: anticipate performance degradation due to input distribution drift
  - **Reliability test**: assess prediction confidence by uncertainty quantification
  - **Robustness test**: assess performance degradation due to small input perturbation
- **Weakness identification**: identify regions and drivers where the model performs weak:
  - Underfitting/overfitting due to nonlinearity or interaction → detectable and fixable
  - Unreliable regions with larger prediction uncertainty → detectable and explainable
  - Sparse data or low signal-to-noise ratio → difficult to predict, potential model limitation
- **Bias and Fairness**: identify disparity or discrimination against demographic groups and mitigate bias.

Model Diagnostics Trilogy (in this UNCC-MDS class)

Error and Resilience	• Accuracy and Residuals, Error Slicing, Underfit and Overfit • Resilience Test, Distribution Drift, Performance Degradation • Segmented Diagnostics, Performance Heterogeneity
Prediction Uncertainty	• Probability calibration for binary classification models • Split conformal prediction for regression models • Identify/explain regions with prediction uncertainty
Bias and Fairness	• Measure of Fairness • Segmented Metrics • Bias Mitigation Methods

Model Diagnostics Part 1: Error and Resilience

- **Data and Model Pipelines**
- Error Analysis
  - Accuracy and Residuals
  - Error Slicing
  - Underfitting and Overfitting
- Resilience Test
  - Performance Degradation
  - Distribution Drift Measurement
- Segmented Diagnostics
  - Data Clustering
  - Performance Heterogeneity

An integrated Python toolbox for interpretable machine learning

```
pip install PiML
```

PiML Package: https://github.com/SelfExplainML/PiML-Toolbox

PiML User Guide: https://selfexplainml.github.io/PiML-Toolbox

Google Colab Notebooks:

- CaliforniaHousing Case (Regression)
- SimuCredit Case (Binary Classification)

Medium PiML Tutorials:

- 10/9/2023: Model Diagnostics Trilogy - Part 1

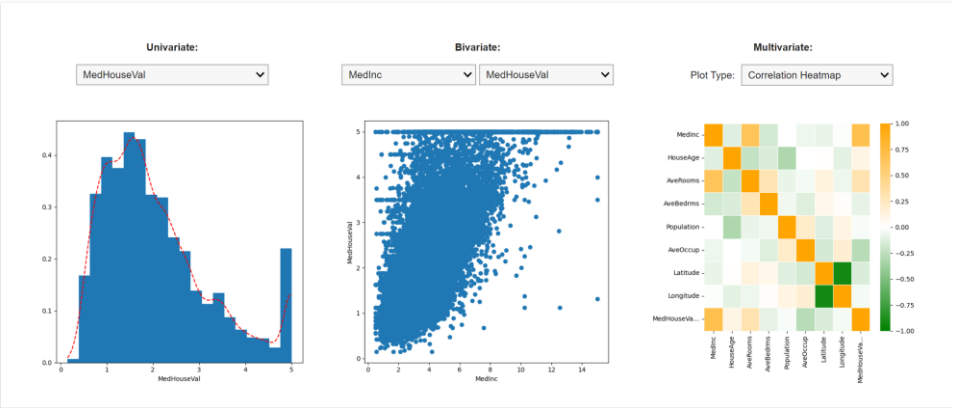
Data and Model Pipelines

- CaliforniaHousing Case (Regression)

```
from piml import Experiment
exp = Experiment()
exp.data_loader(data='CaliforniaHousing_trim2')
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422
...	...	...	...	...	...	...	...	...	...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09	0.781
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21	0.771
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22	0.923
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32	0.847
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24	0.894

20640 rows × 9 columns

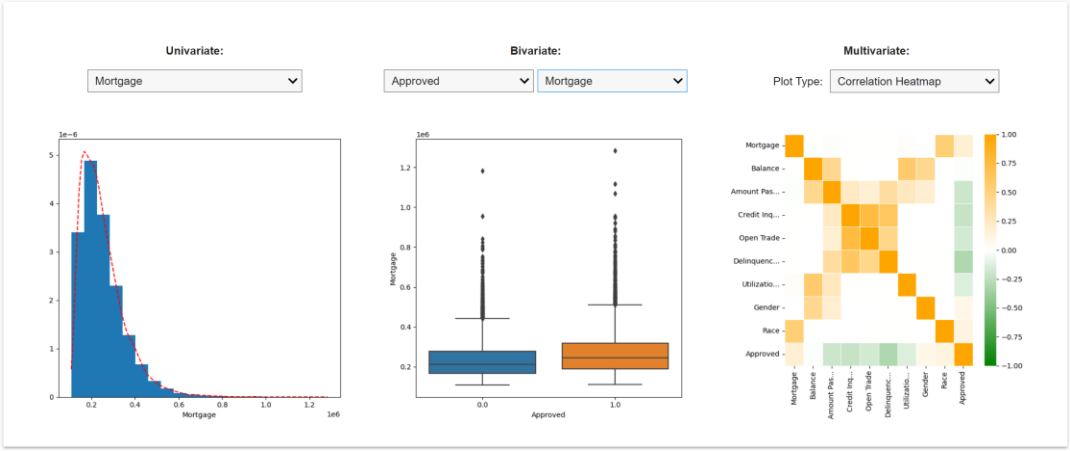


- SimuCredit Case (Binary Classification)

```
from piml import Experiment
exp = Experiment()
exp.data_loader(data='SimuCredit')
```

	Mortgage	Balance	Amount Past Due	Credit Inquiry	Open Trade	Delinquency	Utilization	Gender	Race	Approved
0	196153.90	2115.19	0.00	0.0	0.0	0.0	0.759069	1.0	0.0	1.0
1	149717.49	2713.77	1460.57	1.0	1.0	1.0	0.402820	1.0	0.0	1.0
2	292626.34	2209.01	0.00	0.0	0.0	0.0	0.684272	1.0	1.0	1.0
3	264812.52	21.68	0.00	0.0	0.0	0.0	0.037982	0.0	0.0	0.0
4	236374.39	1421.49	1290.85	0.0	0.0	2.0	0.231110	1.0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...
19995	236123.54	3572.34	0.00	0.0	0.0	0.0	0.896326	1.0	1.0	0.0
19996	374572.72	3560.24	0.00	0.0	0.0	0.0	0.648893	1.0	1.0	0.0
19997	279238.55	101.75	0.00	0.0	0.0	0.0	0.068079	0.0	1.0	0.0
19998	149678.27	439.46	214.36	1.0	0.0	2.0	0.311219	0.0	0.0	1.0
19999	265153.92	909.82	0.00	0.0	0.0	0.0	0.300862	1.0	1.0	1.0

20000 rows × 10 columns



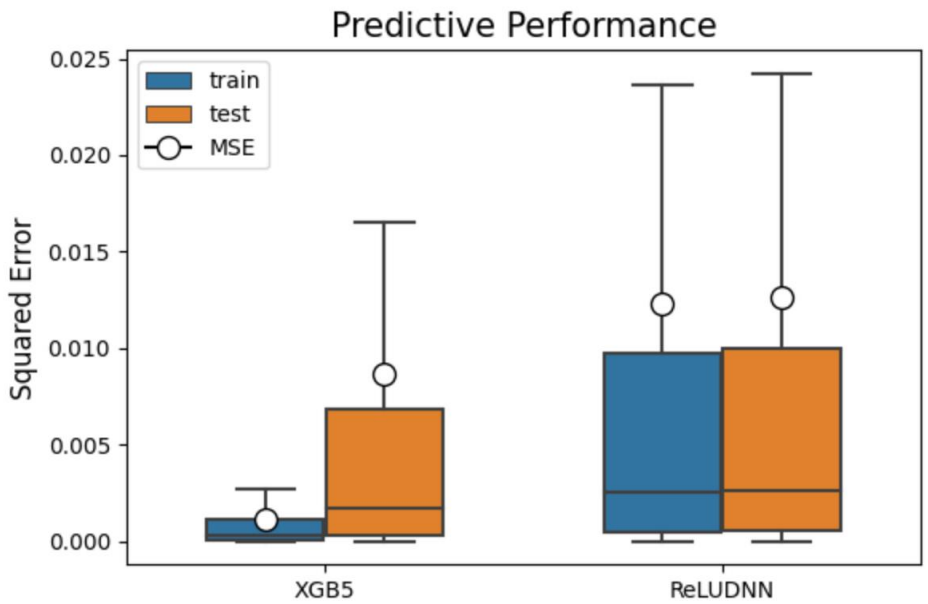
Data and Model Pipelines

- CaliforniaHousing Case (Regression)

```
from xgboost import XGBRegressor
XGB5 = XGBRegressor(max_depth=5, n_estimators=500)
exp.model_train(model = XGB5, name='XGB5')
```

```
from sklearn.neural_network import MLPRegressor
ReLUDNN = MLPRegressor(hidden_layer_sizes=[40]*4, activation="relu",
                       random_state=0)
exp.model_train(model = ReLUDNN, name='ReLUDNN')
```

```
exp.model_compare(models=["XGB5", "ReLUDNN"], show="accuracy_plot",
                  metric="MSE", figsize=(6, 4))
```

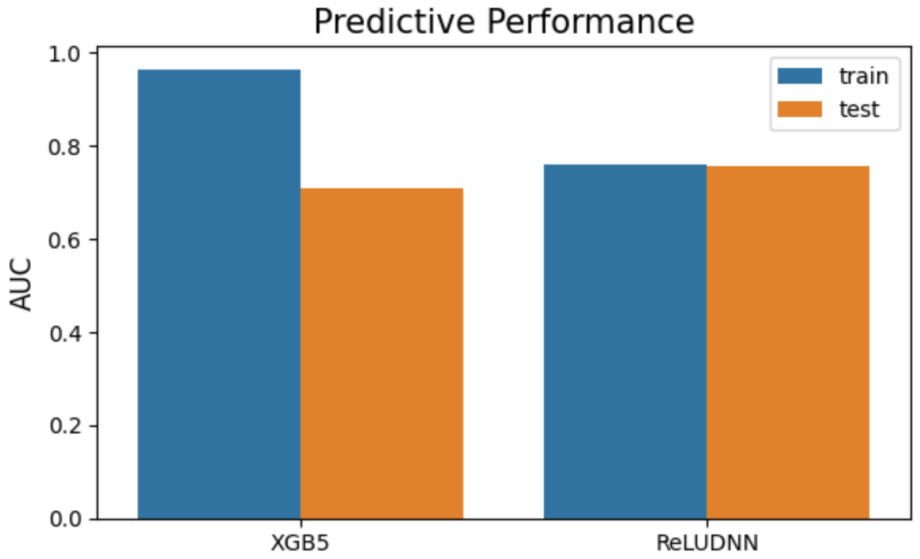


- SimuCredit Case (Binary Classification)

```
from xgboost import XGBClassifier
XGB5 = XGBClassifier(max_depth=5, n_estimators=500)
exp.model_train(model = XGB5, name='XGB5')
```

```
from sklearn.neural_network import MLPClassifier
ReLUDNN = MLPClassifier(hidden_layer_sizes=[20]*2, activation="relu",
                        random_state=0)
exp.model_train(model = ReLUDNN, name='ReLUDNN')
```

```
exp.model_compare(models=["XGB5", "ReLUDNN"], show="accuracy_plot",
                  metric="AUC", figsize=(6, 4))
```

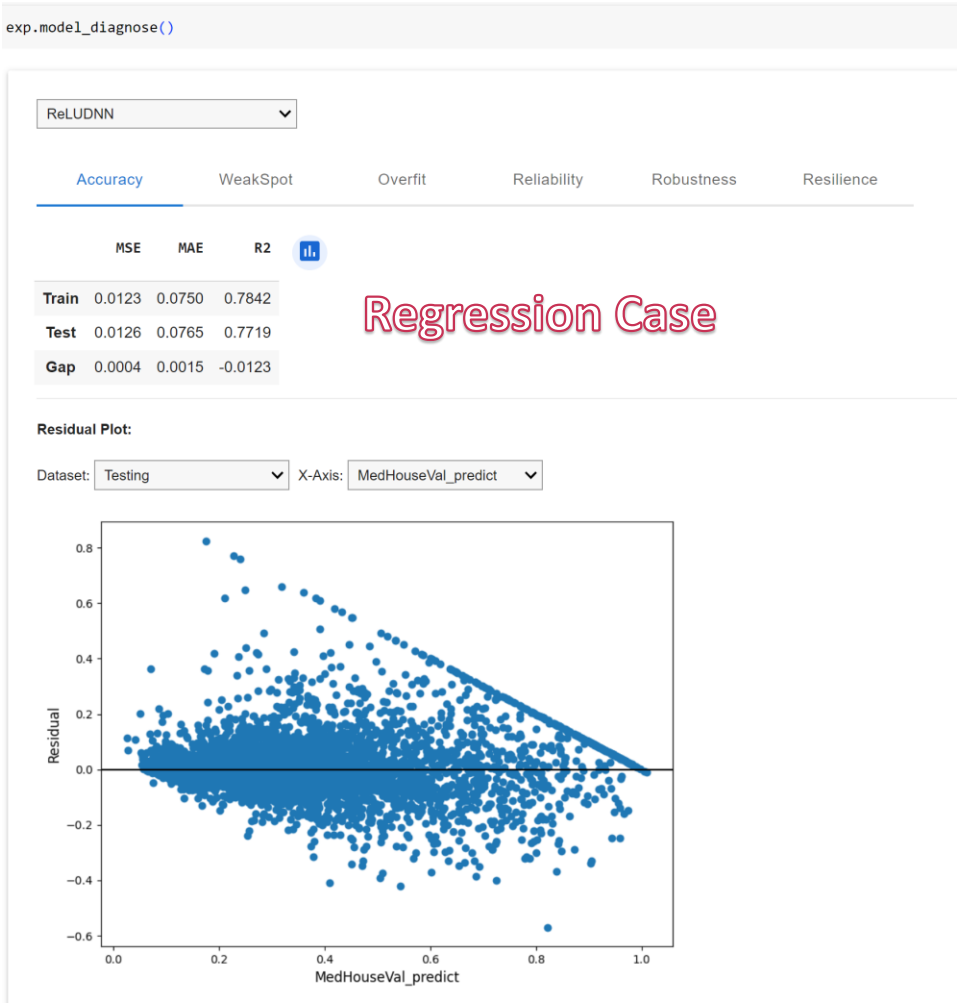


Model Diagnostics Part 1: Error and Resilience

- Data and Model Pipelines
- **Error Analysis**
  - Accuracy and Residuals
  - Error Slicing
  - Underfitting and Overfitting
- Resilience Test
  - Performance Degradation
  - Distribution Drift Measurement
- Segmented Diagnostics
  - Data Clustering
  - Performance Heterogeneity

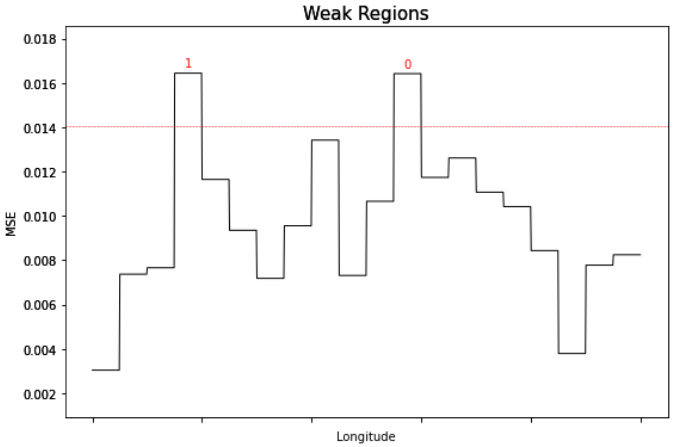
Accuracy and Residuals

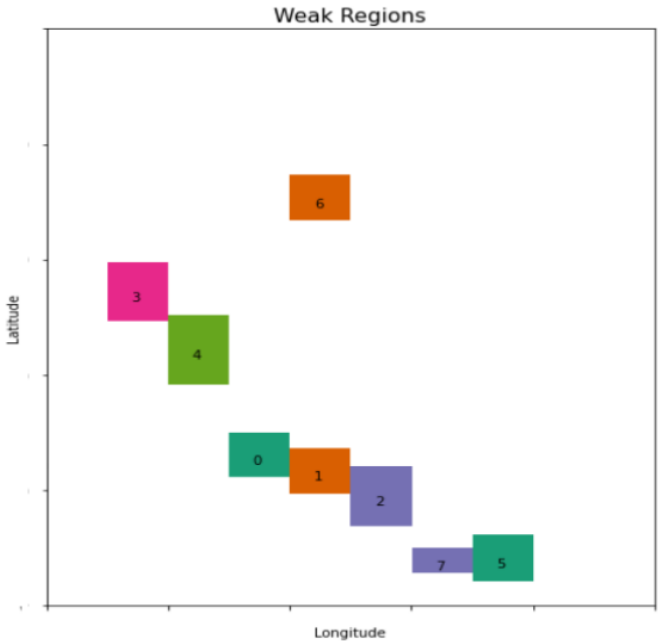
- As usual, regression and classification models can be assessed by accuracy and residuals.



Error Slicing Techniques

1. **Specify an appropriate metric** based on individual prediction residuals: e.g., MSE for regression, ACC for classification, train-test performance gap, prediction interval bandwidth, etc.
2. Specify 1 or 2 slicing features of interest;
3. Evaluate the metric for each sample in the target data (training or testing) as pseudo responses;
4. Segment the target data along the slicing features, by
   a) [Unsupervised] Histogram slicing with equal-space binning, or
   b) [Supervised] fitting a decision tree or tree-ensemble to generate the sub-regions;
5. **Identify the sub-regions** with average metric exceeding the pre-specified threshold, subject to minimum sample condition.

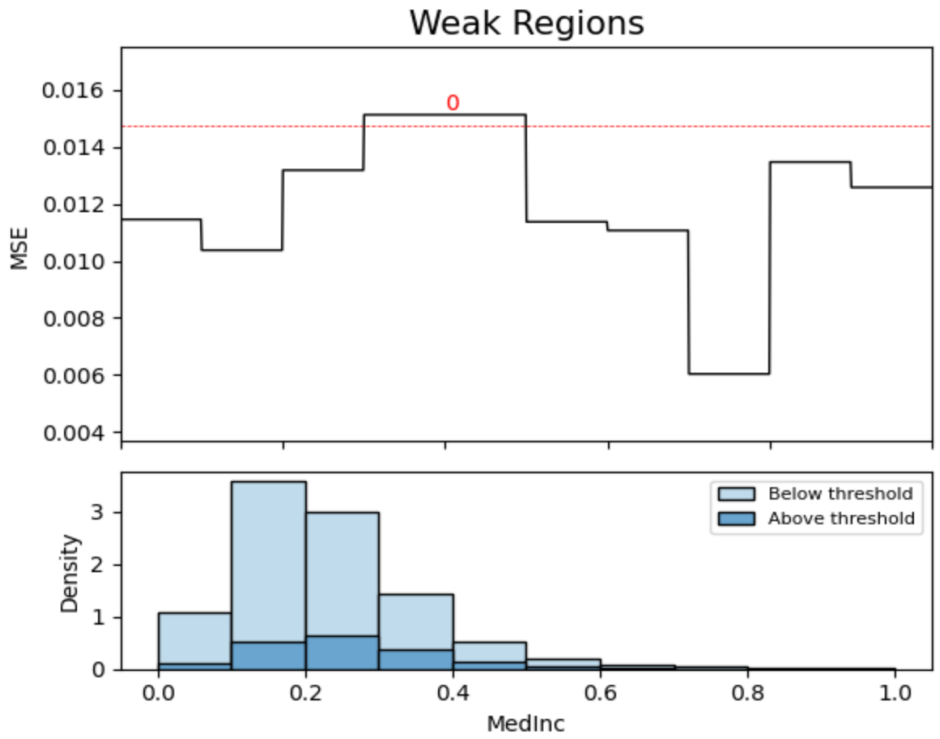




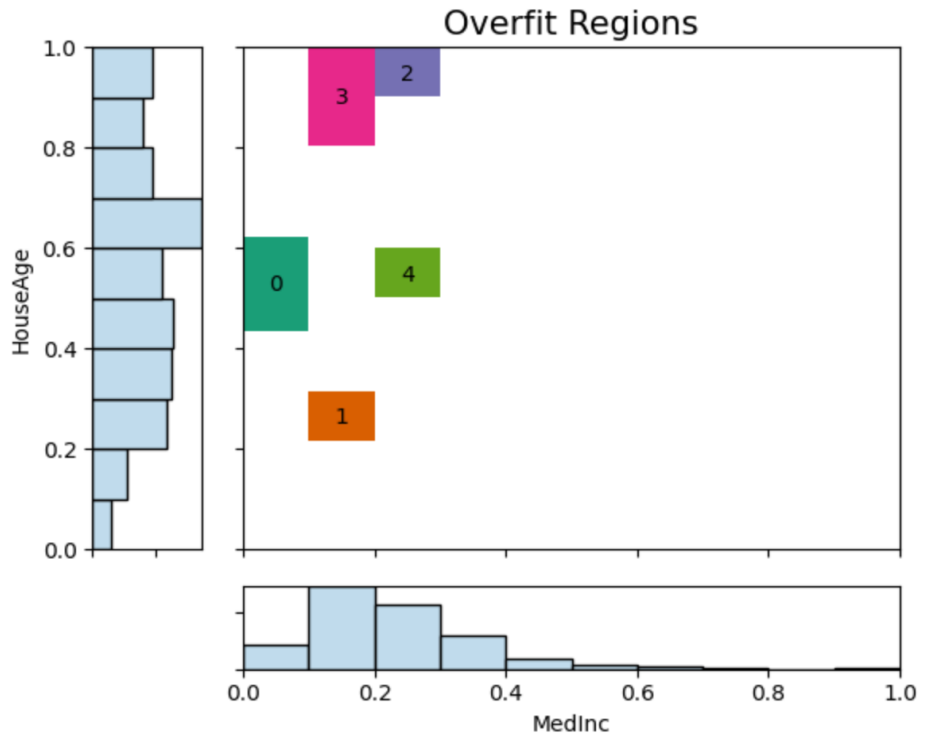
Underfitting and Overfitting

- Apply error slicing to identify regions with underfitting/overfitting weakness

```
exp.model_diagnose(model="ReLUDNN", show="weakspot", metric="MSE",
                   slice_method="histogram", slice_features=["MedInc"],
                   threshold=1.2, min_samples=20, use_test=False, figsize=(6,5))
```




```
exp.model_diagnose(model="ReLUDNN", show="overfit", metric="MSE",
                   slice_method="histogram", slice_features=["MedInc", "HouseAge"],
                   threshold=1.2, min_samples=100, figsize=(6, 5))
```



Model Diagnostics Part 1: Error and Resilience

- Data and Model Pipelines
- Error Analysis
  - Accuracy and Residuals
  - Error Slicing
  - Underfitting and Overfitting
- **Resilience Test**
  - Performance Degradation
  - Distribution Drift Measurement
- Segmented Diagnostics
  - Data Clustering
  - Performance Heterogeneity

Resilience Test

- Resilience test is to anticipate performance degradation under covariate distribution drift.

$$P_{train}(Y|X) = Q_{use}(Y|X) \text{ but } P_{train}(X) \neq Q_{use}(X)$$

- Distributionally resilient models would show mild degradation in performance under distribution drift.

$$\min_{\theta \in \Omega} \max_{Q \in \mathcal{S}} \mathbb{E}_Q[l(X, Y; \theta)], \text{ where } \mathcal{S} = \{\text{Distribution drift scenarios}\}$$

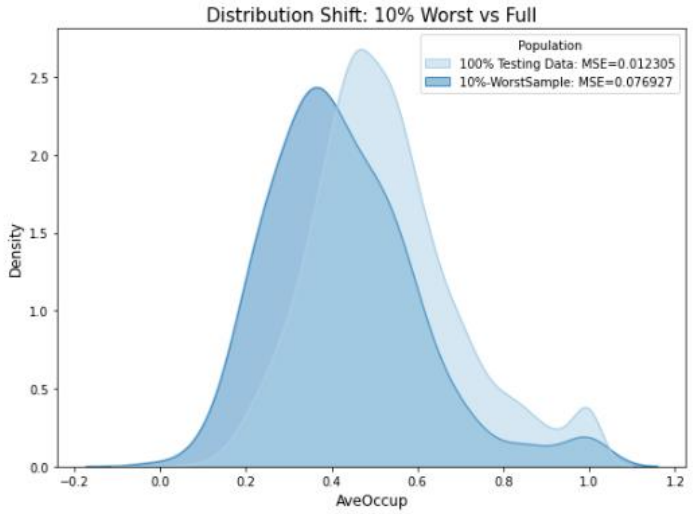
- **Resilient scenarios** offered in the PiML toolbox:
  1. Worst-sample: percentage of worst-performing test samples based on residual magnitude
  2. Worst-cluster: worst-performing cluster of test samples based on K-means clustering
  3. Outer-sample: percentage of boundary/outlying test samples distant from the sample mean
  4. Hard-sample: percentage of difficult-to-predict test samples based on an auxiliary model
- Note that scenarios 1 and 2 are model-specific, while scenarios 3 and 4 are model-agnostic.

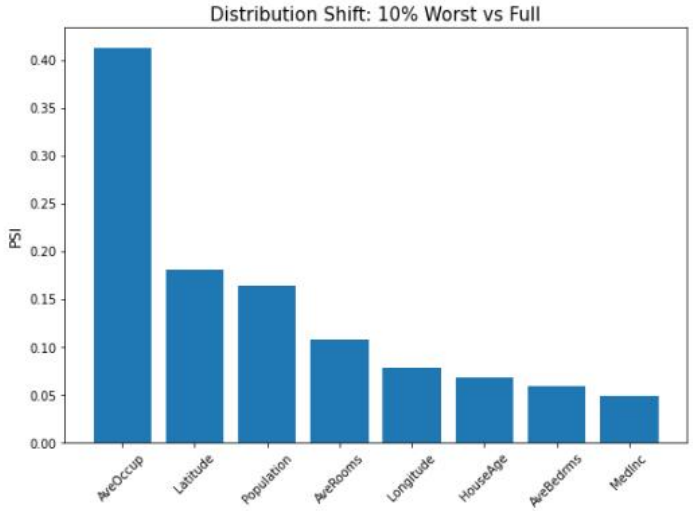
Distribution Drift Measurement

- **Two-sample test** between empirical distributions**:**
  - Kullback-Leibler (KL) divergence, Kolmogorov-Smirnov (KS) and Cramer-von Mises (CM) statistics, Population Stability Index (PSI), Wasserstein distance
- **Population Stability Index:** one-variable-at-a-time measurement

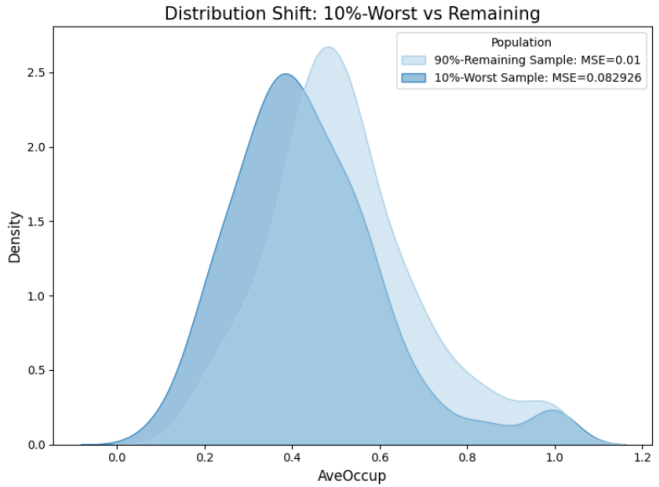
$$PSI = \sum_{i=1}^{B} (\text{Target}_i\% - \text{Base}_i\%) \ln\left(\frac{\text{Target}_i\%}{\text{Base}_i\%}\right)$$

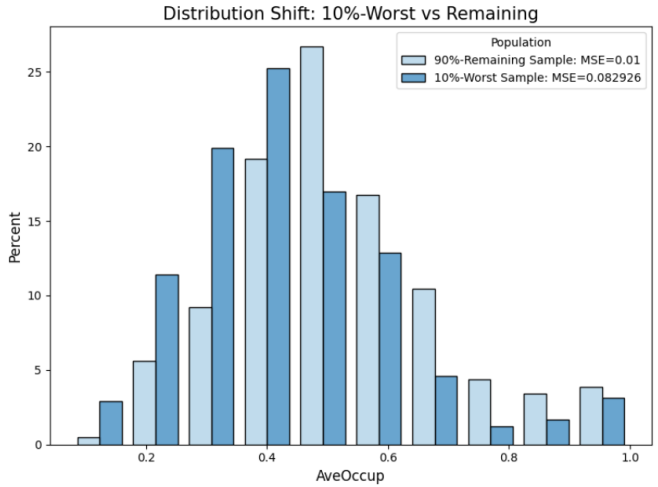
  based on the proportions of samples in each bucket of the target vs. base population. Rule of thumb:
  - PSI < 0.1: no significant distribution change
  - PSI < 0.2: moderate distribution change
  - PSI >= 0.2: significant distribution change
- Variables with notable drift are deemed to be sensitive or vulnerable in the resilience test. They can be further verified through WeakSpot error slicing.

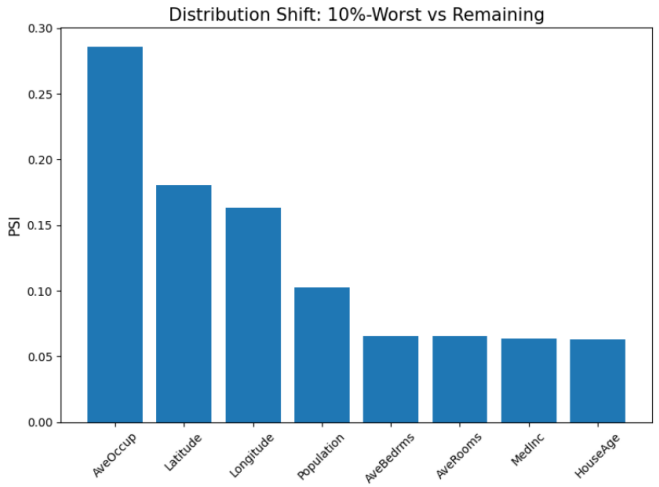


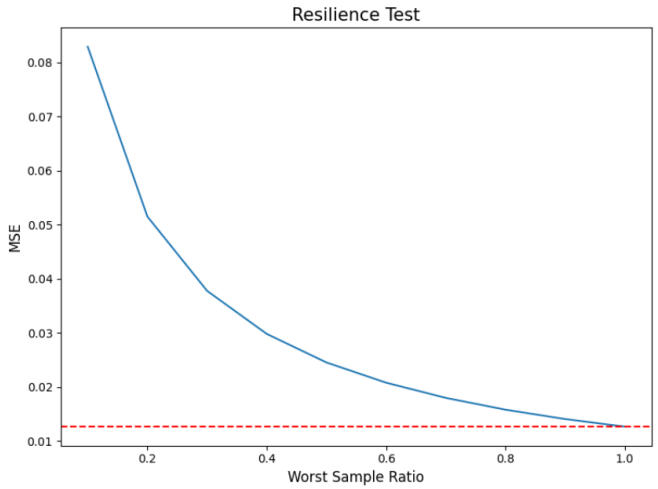


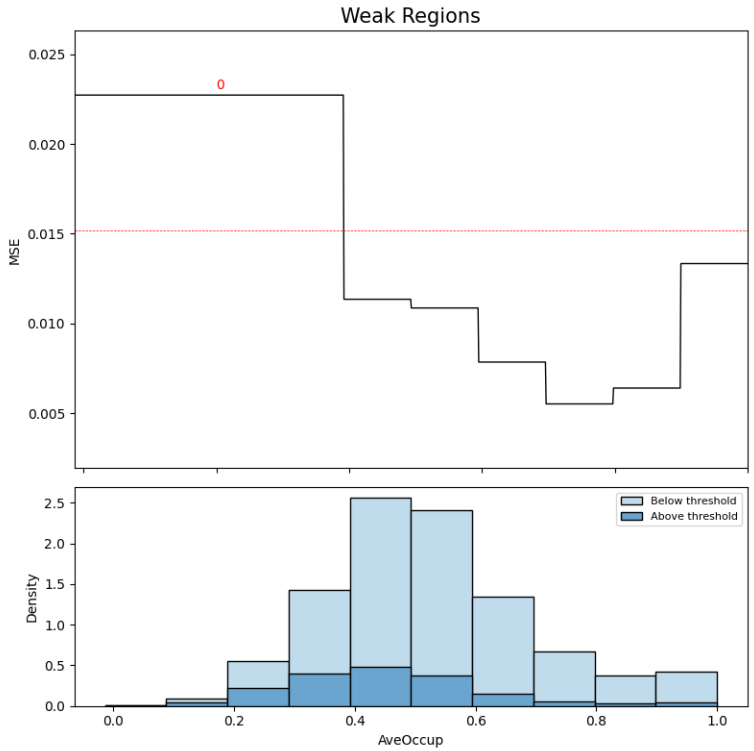
Resilience Test: CaliforniaHousing-ReLUDNN Case



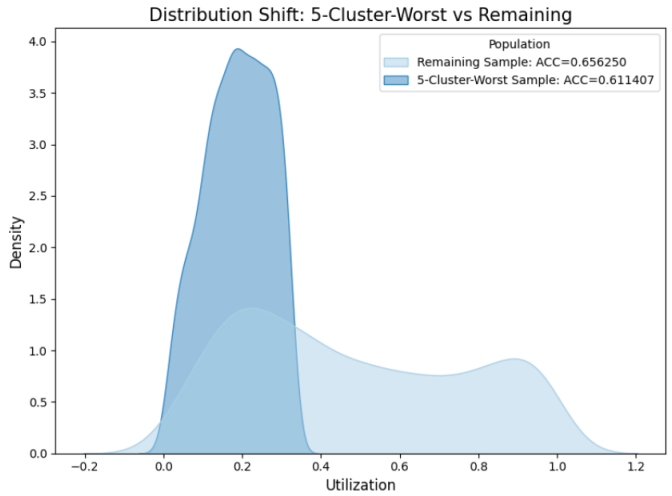


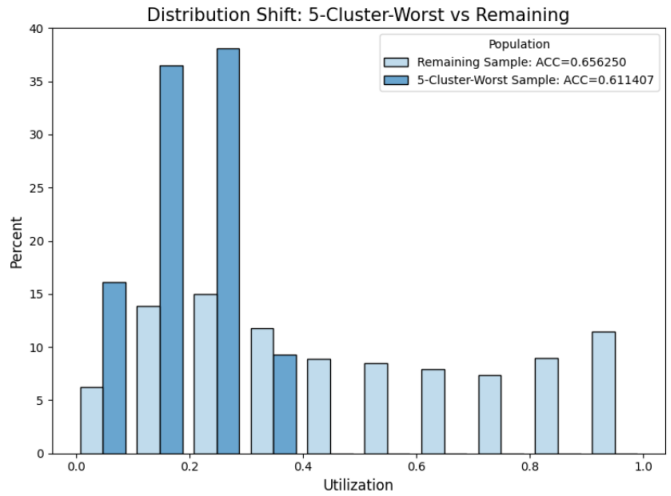


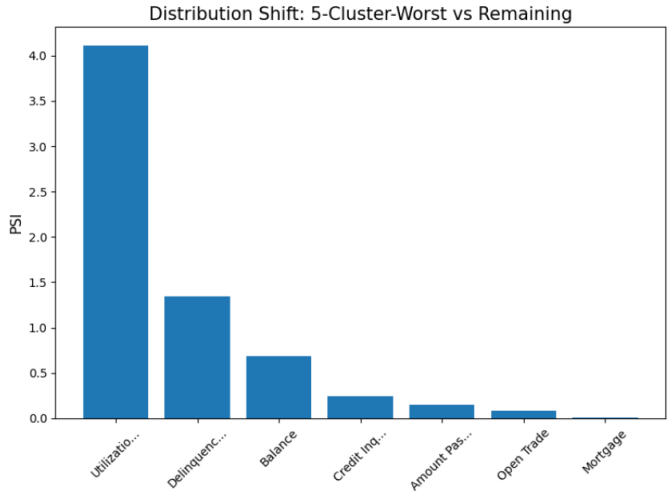


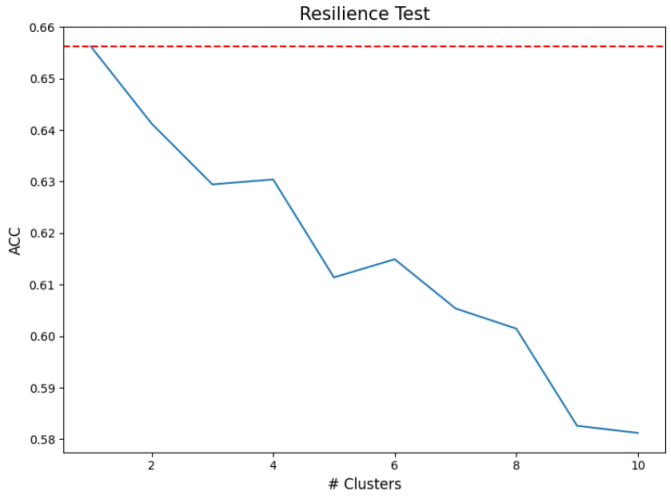


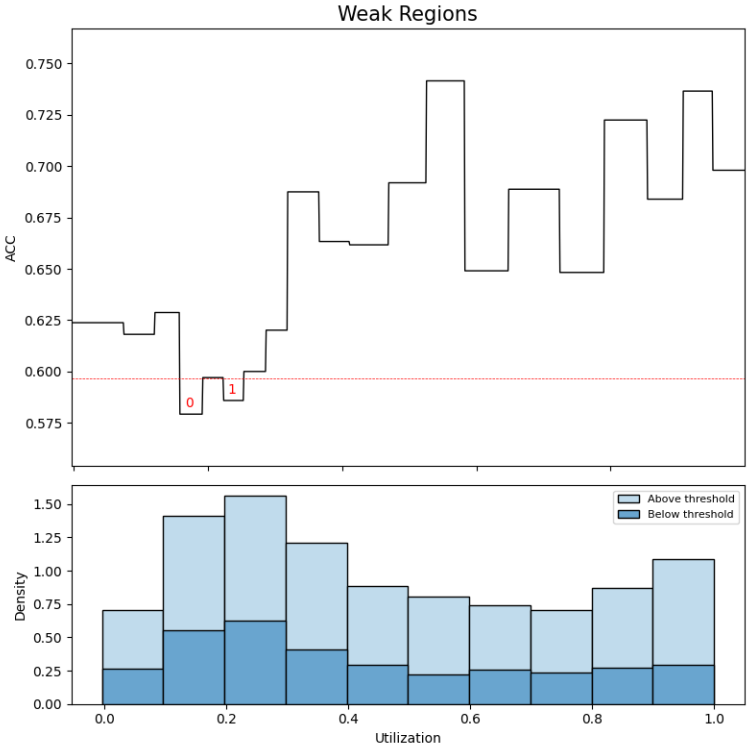
Resilience Test: SimuCredit-XGB5 Case











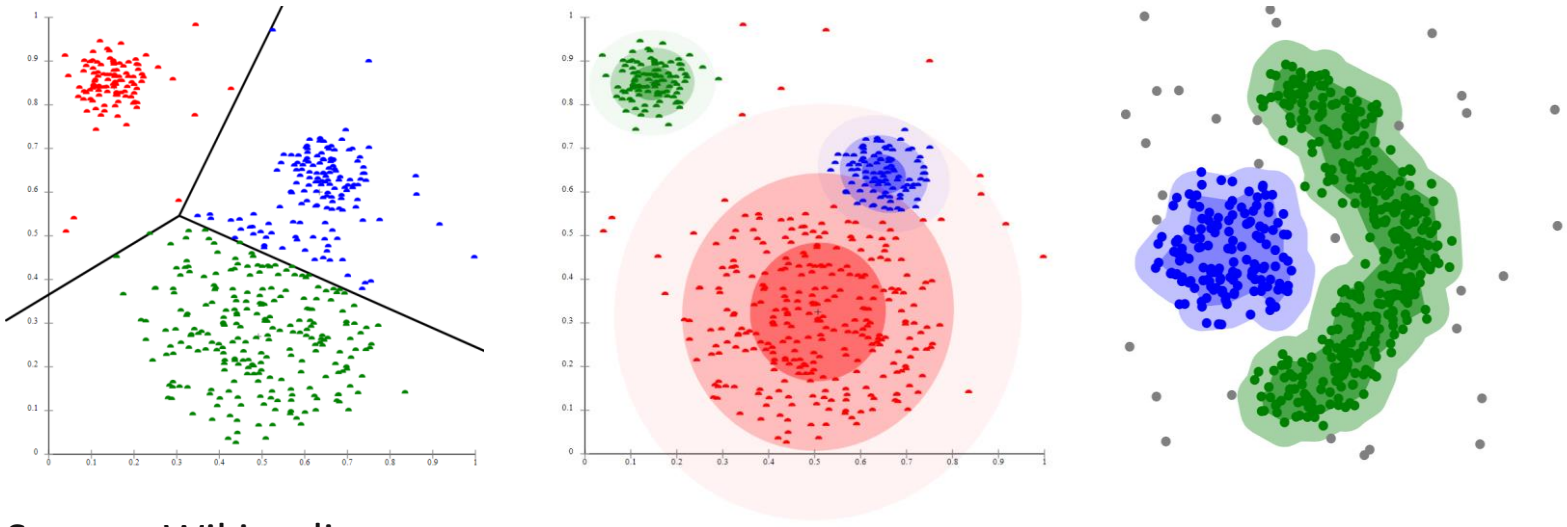
Model Diagnostics Part 1: Error and Resilience

- Data and Model Pipelines
- Error Analysis
  - Accuracy and Residuals
  - Error Slicing
  - Underfitting and Overfitting
- Resilience Test
  - Performance Degradation
  - Distribution Drift Measurement
- **Segmented Diagnostics**
  - Data Clustering
  - Performance Heterogeneity

Segmented Diagnostics

- In the spirit of resilience test under worst-cluster scenario, model diagnostics can be conducted on a segment-by-segment basis.

- **Data clustering algorithms**: K-Means, Gaussian Mixture Model, DBSCAN, Hierarchical Clustering, …



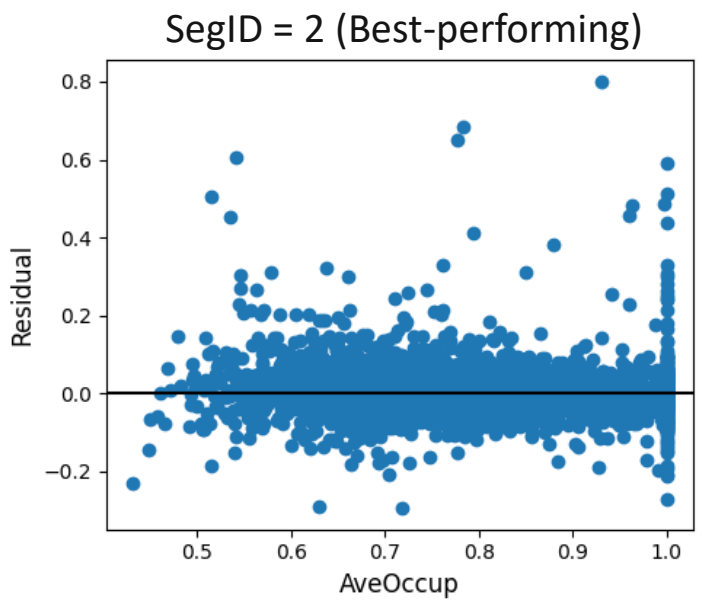
Source: Wikipedia

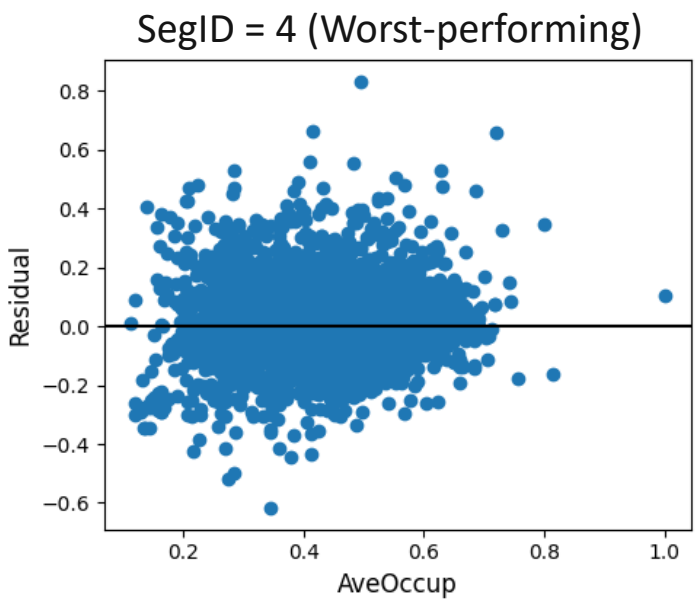
Performance Heterogeneity

- For CaliforniaHousing regression case, run K-Means with K= 6, then conduct segmented error analysis by using PiML scored_test APIs (latest release in V0.5.1):

```
from piml.scored_test import test_accuracy, residual_plot, slicing_weakspot
```

	MSE
0	0.0155
1	0.0135
2	0.0077
3	0.0082
4	0.0178
5	0.0132





- Such experimental “segmented diagnostics” may appear as a new feature in future PiML release.

WELLS FARGO

Thank you

Aijun Zhang, Ph.D.

Email: Aijun.Zhang@wellsfargo.com

LinkedIn: https://www.linkedin.com/in/ajzhang/

PiML

An integrated Python toolbox for interpretable machine learning

```
pip install PiML
```

PiML Package: https://github.com/SelfExplainML/PiML-Toolbox

PiML User Guide: https://selfexplainml.github.io/PiML-Toolbox

Google Colab Notebooks:

- CaliforniaHousing Case (Regression)
- SimuCredit Case (Binary Classification)

Medium PiML Tutorials:

- 10/9/2023: Model Diagnostics Trilogy - Part 1