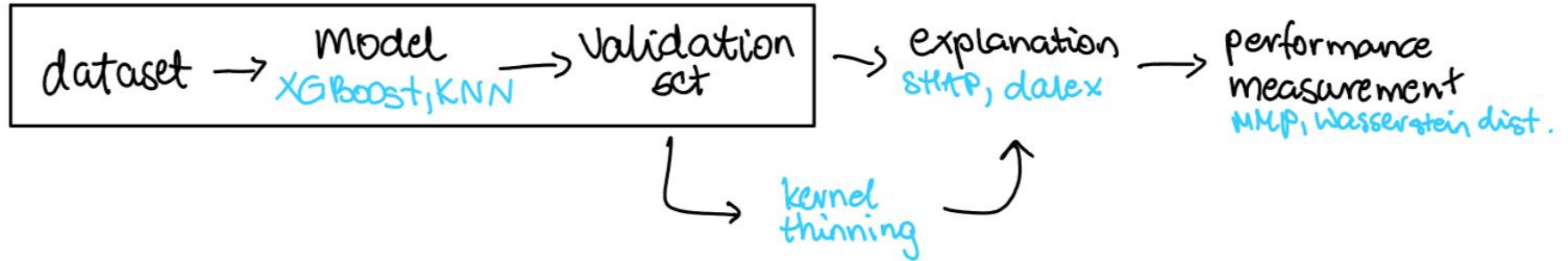# Data compression for improved explanation estimation

Mateusz Biesiadowski, Paulina Kaczyńska, Ania Semik

Pipeline

Code

```python
class Experiment:
    def __init__(self, data_processor, model_class, model_params,
                 shap_class, shap_params, dalex_class, dalex_params,
                 pvi_params, pdp_params, ale_params, pdp_domain=51):
        if pdp_params is None:
            pdp_params = {
                'N': None,
                'verbose': False
            }
        self.data_processor = data_processor
        self.model_class = model_class
        self.model_params = model_params
        self.shap_class = shap_class
        self.shap_params = shap_params
        self.dalex_class = dalex_class
        self.dalex_params = dalex_params
        self.pvi_params = pvi_params
        self.pdp_params = pdp_params
        self.ale_params = ale_params
        self.pdp_domain = pdp_domain
```

```python
def experiment_1(no_tests=10, save_path='./results/exp1_diabetes.parquet', model_metric='r2'):
    X, y = load_diabetes(return_X_y=True)
    data_processor = DataProcessor(X=X, y=y)

    experiment_settings = {
        'data_processor': data_processor,
        'model_class': xgb.XGBRegressor,
        'model_params': {'max_depth': 4, 'subsample': 0.9 ** 3, 'colsample_bytree': 0.9, 'colsample_bylevel': 0.9,
                         'colsample_bynode': 0.9, 'alpha': 0.1},
        'shap_class': shap.explainers.Tree,
        'shap_params': {'model_output': "raw"},
        'dalex_class': dx.Explainer,
        'dalex_params': {'verbose': VERBOSE},
        'pvi_params': {'N': None, 'verbose': VERBOSE},
        'pdp_params': {'N': None, 'verbose': VERBOSE},
        'ale_params': {'type': "accumulated", 'center': False, 'N': None, 'verbose': VERBOSE}
    }

    experiment = Experiment(**experiment_settings)
    result = experiment.run(no_tests, Experiment.kernel_polynomial, save_path=save_path,
                            test_size=4 ** 3, model_metric=model_metric)

    return result
```
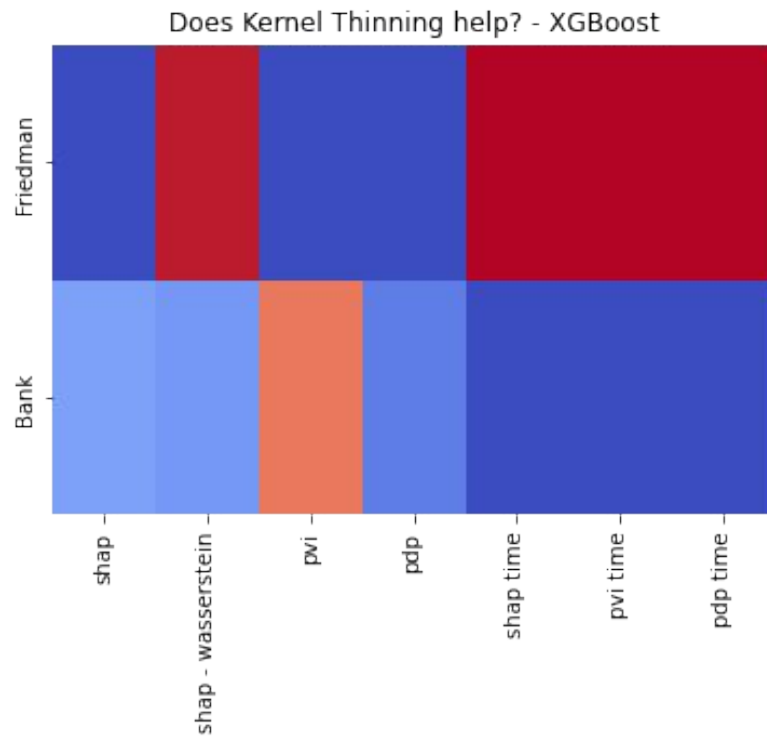
What can we change in experiments?

- Datasets:
  - Friedman dataset (synthetic)
  - Bank Marketing Dataset (45,000)
  - COVID-19 (>1,000,000)
- Models:
  - XGBoost
  - K-nearest neighbors (?)
- Explanation methods:
  - SHAP
  - PVI
  - PDP
  - ALE

Initial observations

- SHAP is so slow that anything helps
- Friedman is too easy currently

Does Kernel Thinning help? - XGBoost

Dziękujemy za uwagę!