ПОЛИТЕХ
Санкт-Петербургский
политехнический университет
Петра Великого

Институт компьютерный наука и технологии (ИКНТ)

COMPUTO ERGO SUM

Решение прикладных задач методами машинного обучения

SCHEMA OF PROJECT

APPLICATION OF SHAP FOR GLOBAL AND LOCAL EXPLANATIONS IN TABULAR DATA

Студент:
Эспинола Ривера, Хольгер Элиас

4 March, 2024

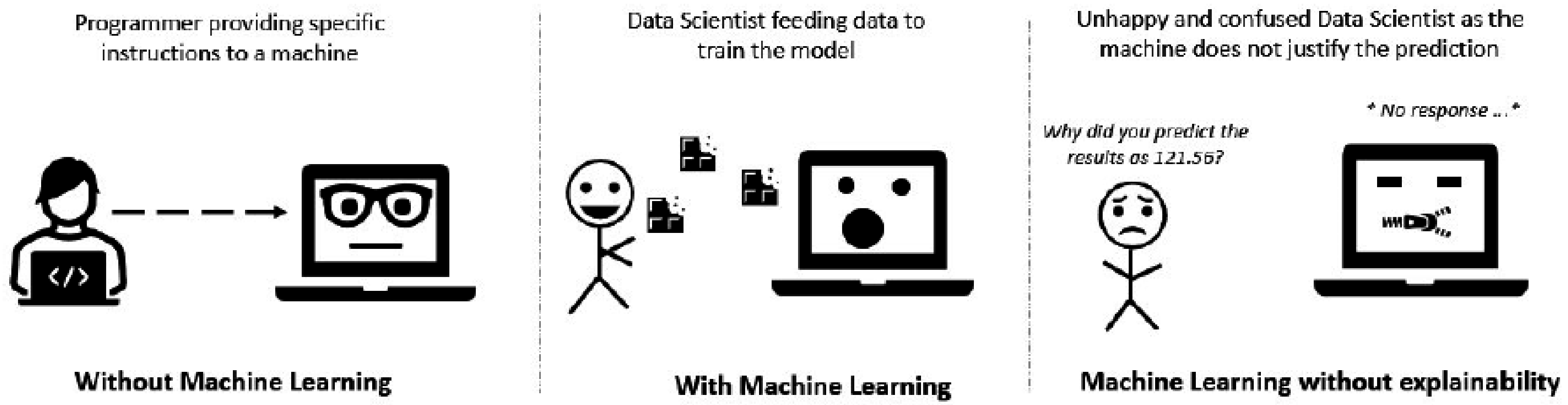
Руковадитель:
Заборовский Владимир Сергеевич

Contents

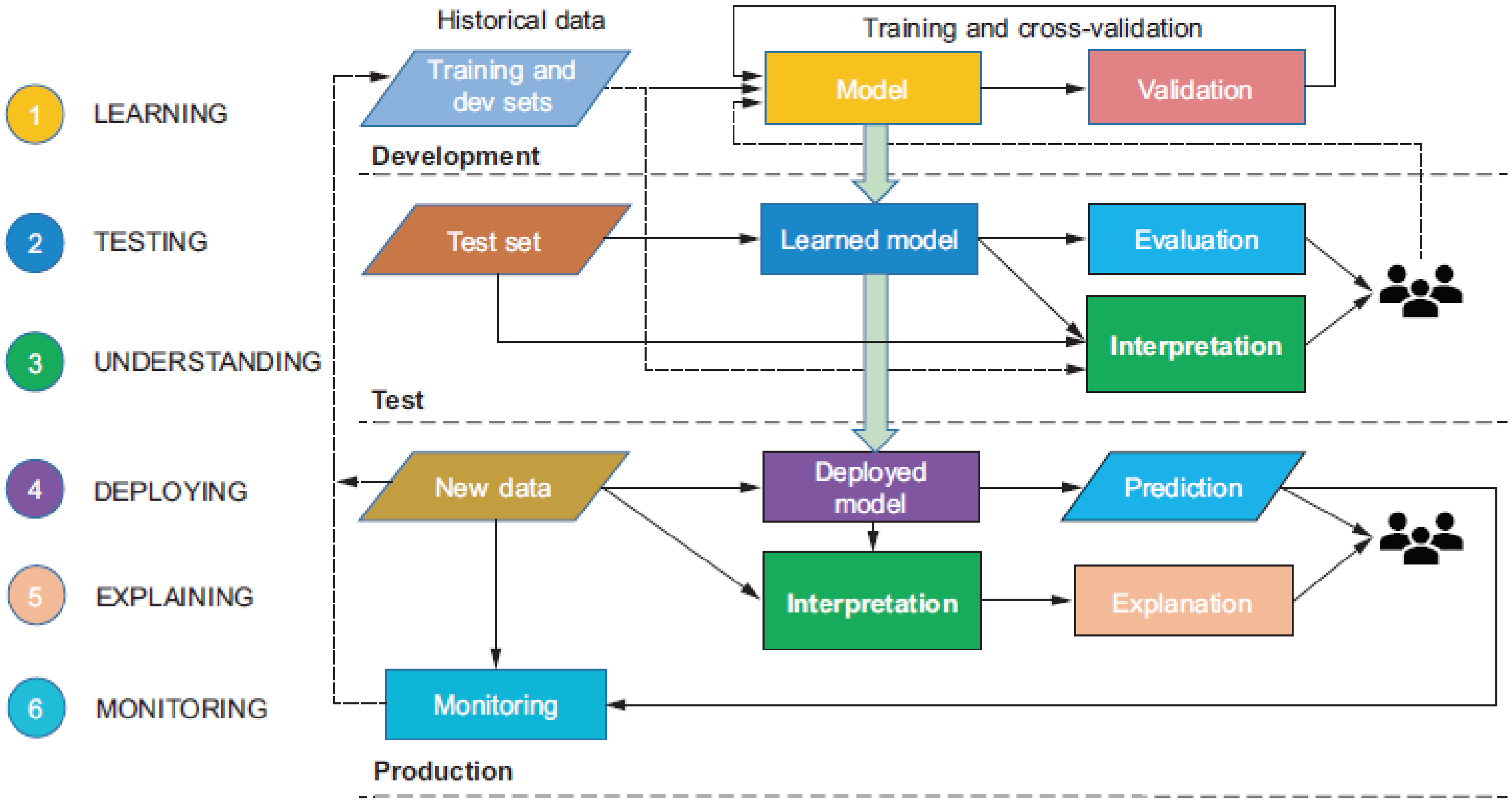
Theoretical basis of X-AI

DEFINITIONS

- Explainable AI (X-AI) is a set of techniques that help interpret and demystify the black-box machine learning models, which needs explainability.
- X-AI provides the visibility under the hood how ML algorithms operates at each stage of solution life cycle.
- X-AI is a methodology that allows to users comprehend **how** ML algorithms take decisions (interpretability) and **why** (explainability).

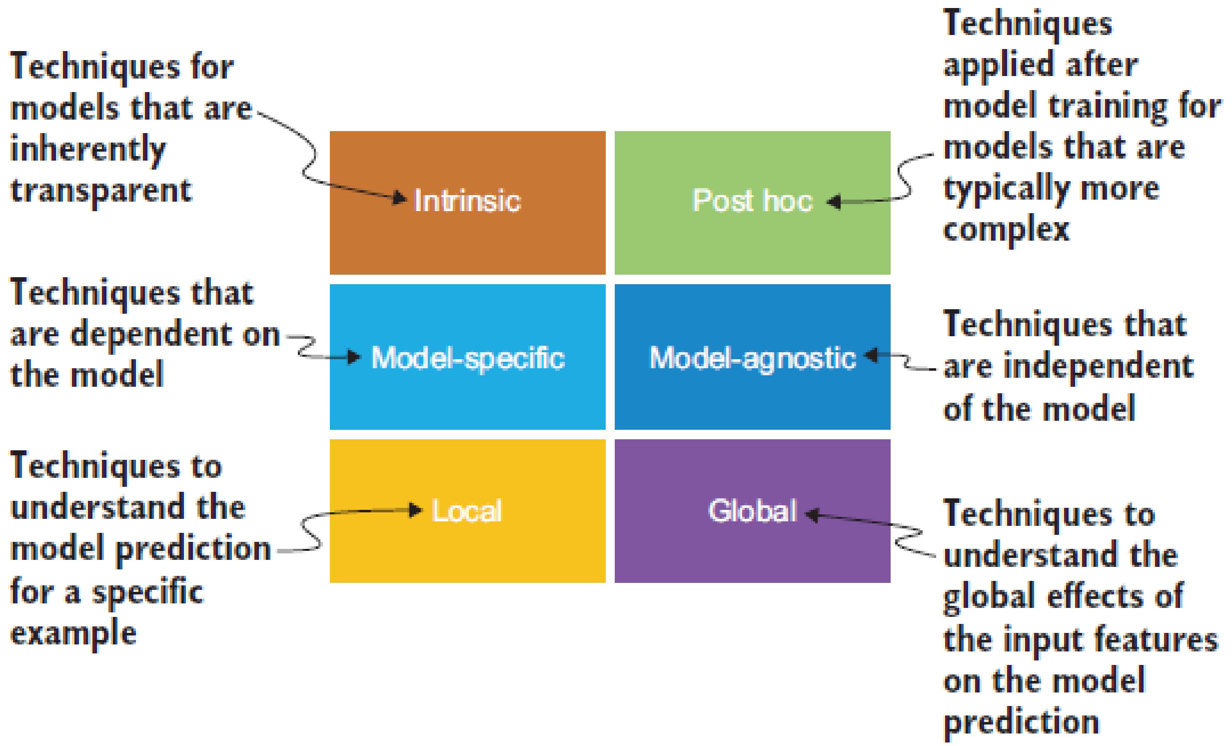


X-AI and building of robust AI-systems

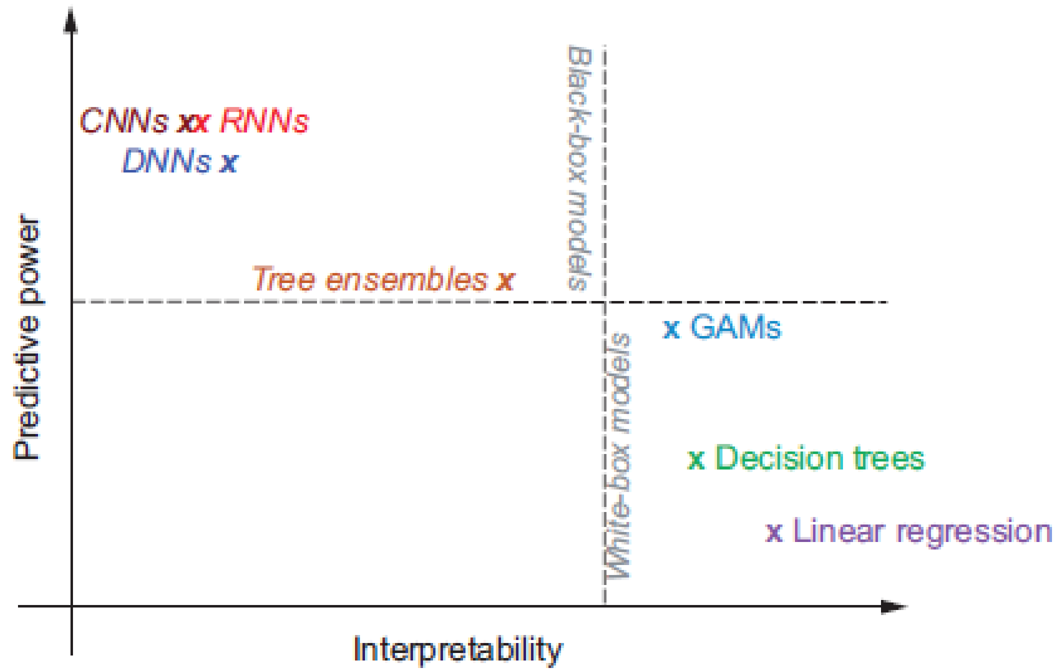


X-AI

Types of Explanation Models



Dilemma interpretability-predictability tradeoff



SHAP Algorithm

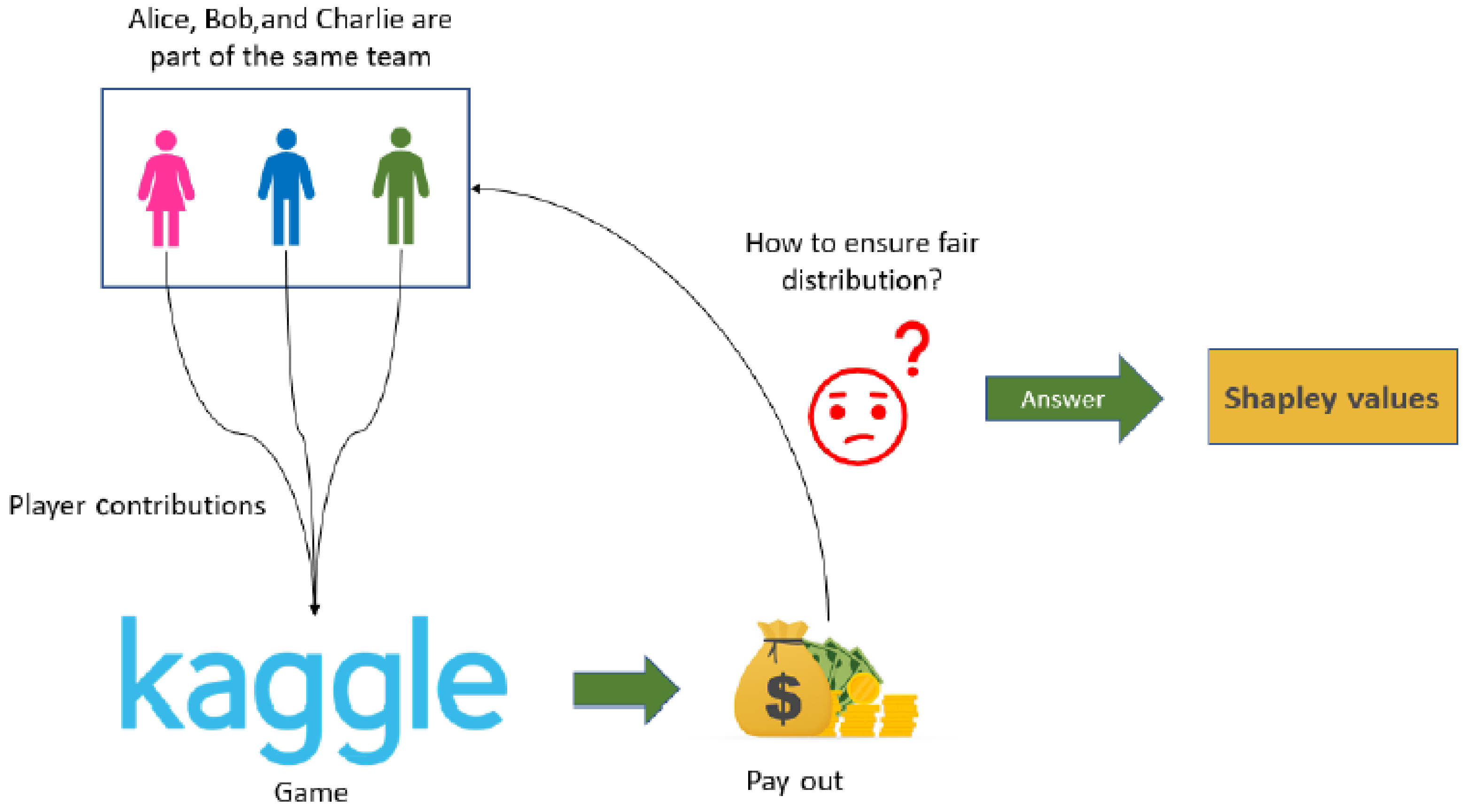
SHAP

SHAP (Shapley Additive Explanations) is a game theory model-agnostic approach to explain the predictions generated by machine learning models.

SHAP is based on **Shapley Values**, used to determine the contribution of each player in a cooperative game.

Intuition about Shapley Values

Consider a team of Kaggle competition conformed by Alice, Bob and Charlie, how can make fair distribution of prize money according player contributions?



Intuition about Shapley Values

Given N players with coalition of S players with v(s) as the total value generated of these S players, the marginal contribution of player **i** is given by:

Where:

$$\underbrace{\varphi(i)}_{\text{Sharpley values}} = \sum_{S\subseteq N/i} \frac{|S|!\,(|N|-|S|-1)!}{|N|!}\left(\underbrace{v(S \cup \{i\})}_{\text{Value accumulated including player } \mathbf{i}} - \overbrace{v(S)}^{\text{Value accumulated without participation of player } \mathbf{i}}\right)$$

Intuition about Shapley Values

Consider the contribution values for all possible combinations of players.

Players	Point Values (V)
Alice	10
Bob	20
Charlie	25
Alice and Bob	40
Alice and Charlie	30
Bob and Charlie	50
Alice, Bob and Charlie	90

Order	Scenario	Contribution V
A, B, C	{A}	V(A) = 10
A, C, B	{A}	V(A) = 10
B, A, C	{A, B}	V(A, B) – V(B) = 40 – 20 = 20
C, A, B	{A, C}	V(A, C) – V(C) = 30 – 25= 5
B, C, A	{A, B, C}	V(A, B, C) – V(B, C) = 90 – 50 = 40
C, B, A	{A, B, C}	V(A, B, C) – V(B, C) = 90 – 50 = 40
Φ(A)	A marg. contribution	$\frac{1}{6}(10 + 10 + 20 + 5 + 40 + 40) = 20.83$

Intuition about Shapley Values

Order	Scenario	Contribution V
B, A, C	{B}	V(B) = 20
B, C, A	{B}	V(A) = 20
A, B, C	{A, B}	V(A, B) – V(A) = 40 – 10 = 30
C, B, A	{B, C}	V(B, C) – V(C) = 50 – 25= 25
A, C, B	{A, B, C}	V(A, B, C) – V(A, C) = 90 – 30 = 60
C, A, B	{A, B, C}	V(A, B, C) – V(A, C) = 90 – 30 = 60
Φ(A)	A marg. contribution	$\frac{1}{6}(20+20+30+25+60+60) = 35.83$

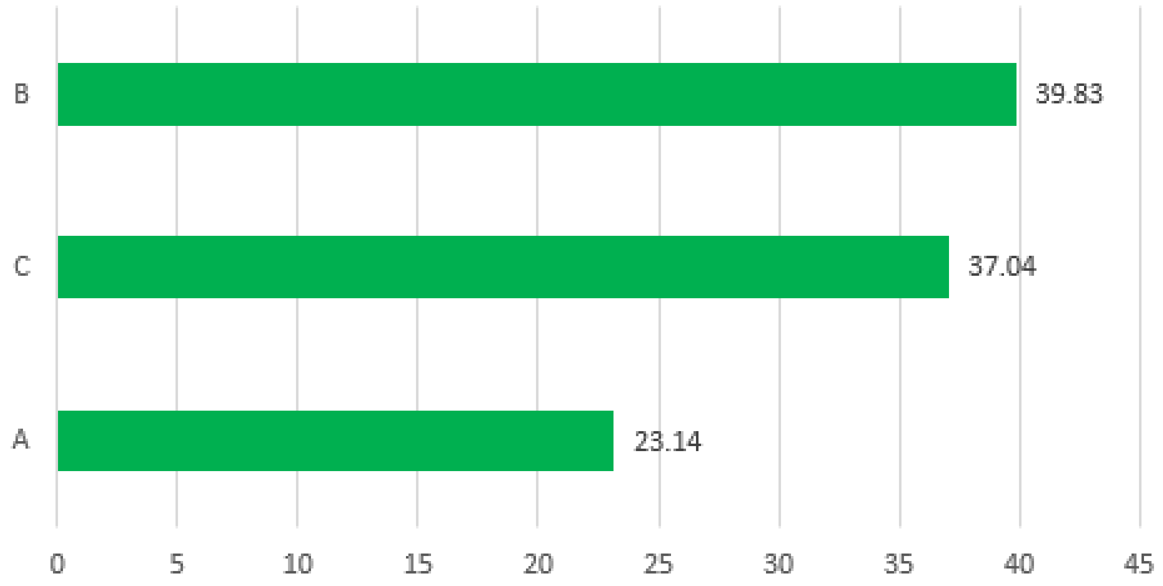
Order	Scenario	Contribution V
C, A, B	{C}	V(C) = 25
C, B, A	{C}	V(C) = 25
A, C, B	{A, C}	V(A, C) – V(A) = 30 – 10 = 20
B, C, A	{B, C}	V(B, C) – V(B) = 50 – 20= 30
A, B, C	{A, B, C}	V(A, B, C) – V(A, B) = 90 – 40 = 50
B, A, C	{A, B, C}	V(A, B, C) – V(A, B) = 90 – 40 = 50
Φ(A)	A marg. contribution	$\frac{1}{6}(25+25+20+30+50+50) = 33.34$

Intuition about Shapley Values

Finally, level of contribution of each player:

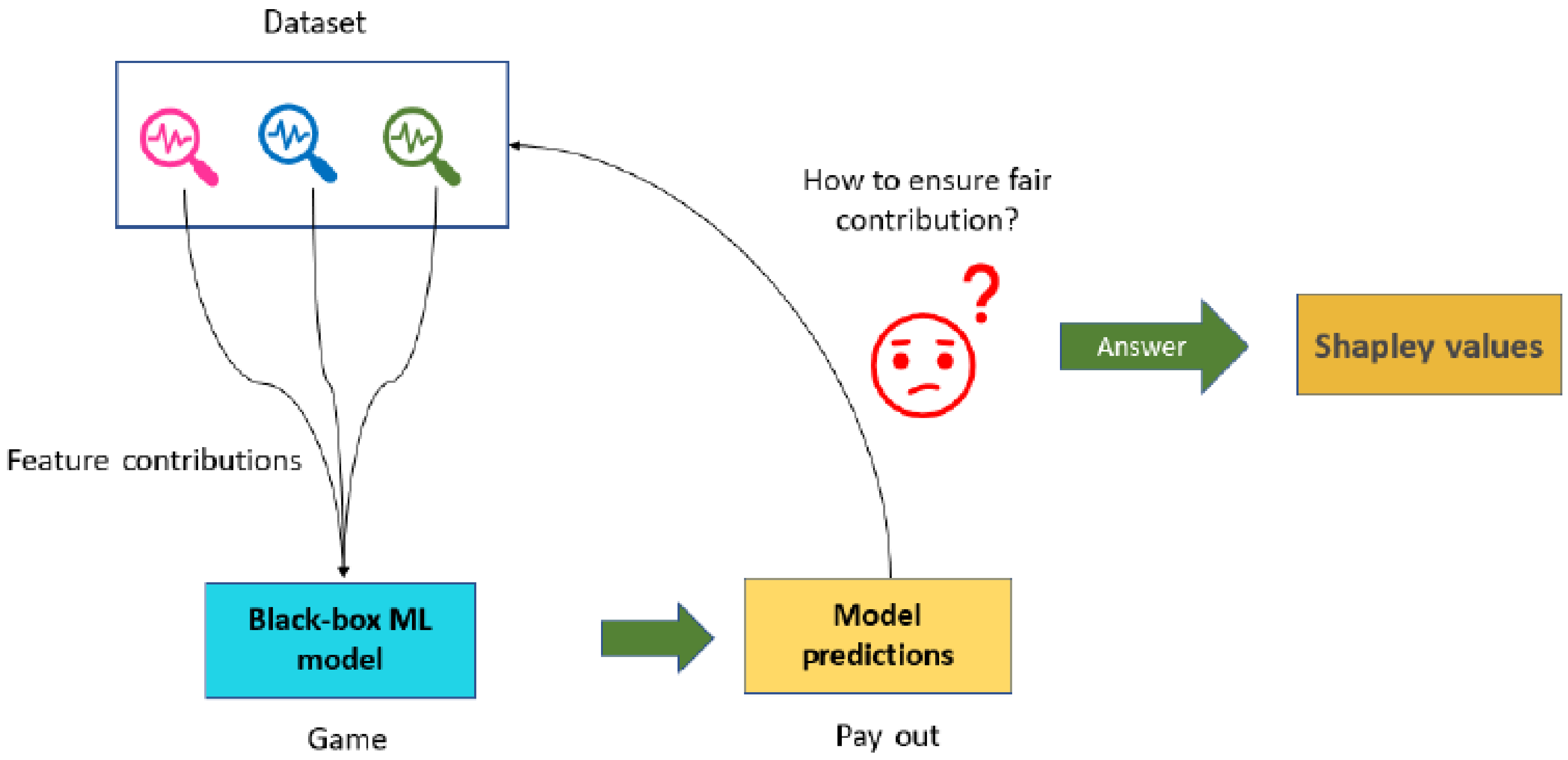
Φ(A)	Φ(B)	Φ(C)	TOTAL
20.83	35.83	33.34	90
23.14%	39.82%	37.04%	100%

Players Contribution according Sharpley values



Shapley Values in context of ML

Shapley values in ML helps to understand the collective contribution of each feature toward the outcome predicted by ML models.



SHAP Algorithms

Global explainability

Local explainability

For aggregating of individual predictions

SHAP

For individual predictions

Model-agnostic

Model-specific

SHAP Explainer: based on Shapley sampling values
Kernel-SHAP: based on LIME approach

Linear SHAP: for linear models
Tree SHAP: for trees and tree-ensembles
Deep SHAP: for deep learning models

Problem Statement

Problem

Task 1: X-AI with SHAP applied to Generated Clusters data

The original data set consists of several clusters generated by different linear laws.

Necessary:

a) For each point in the test sample, determine the vector of feature importances.

b) Cluster the resulting importance vectors.

c) Determine the law by which each of the clusters is generated.

Dataset

Dataset consists in 2 subsets:

- Train set: 1125 samples
- Test set: 375 samples

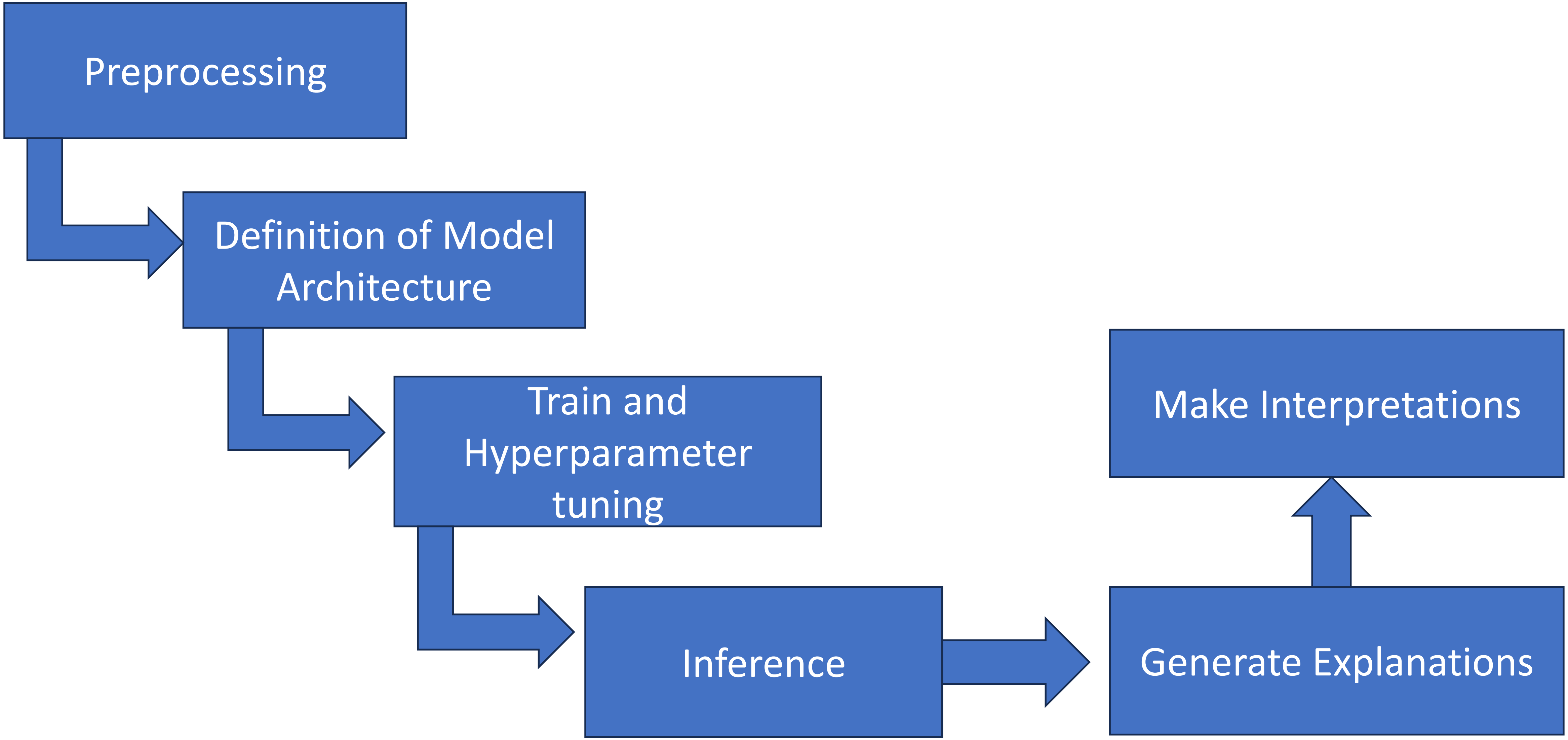
	f0	f1	f2	f3	f4	y
0	1.7005	1.3531	1.2401	2.0728	1.6070	1.6064
1	0.7965	0.7256	0.5263	0.7339	0.8046	0.7234
2	2.8373	2.2479	3.0573	2.9302	2.8407	2.3597
3	3.0394	2.8822	2.4963	2.6470	2.7702	2.8105
4	2.0894	1.2275	1.9977	1.8091	1.1260	1.8924

Dataset have 5 independent variables: f0, f1, f2, f3, f4 [float datatype]
Dataset have 1 real dependent variable: y [float datatype]

Task:

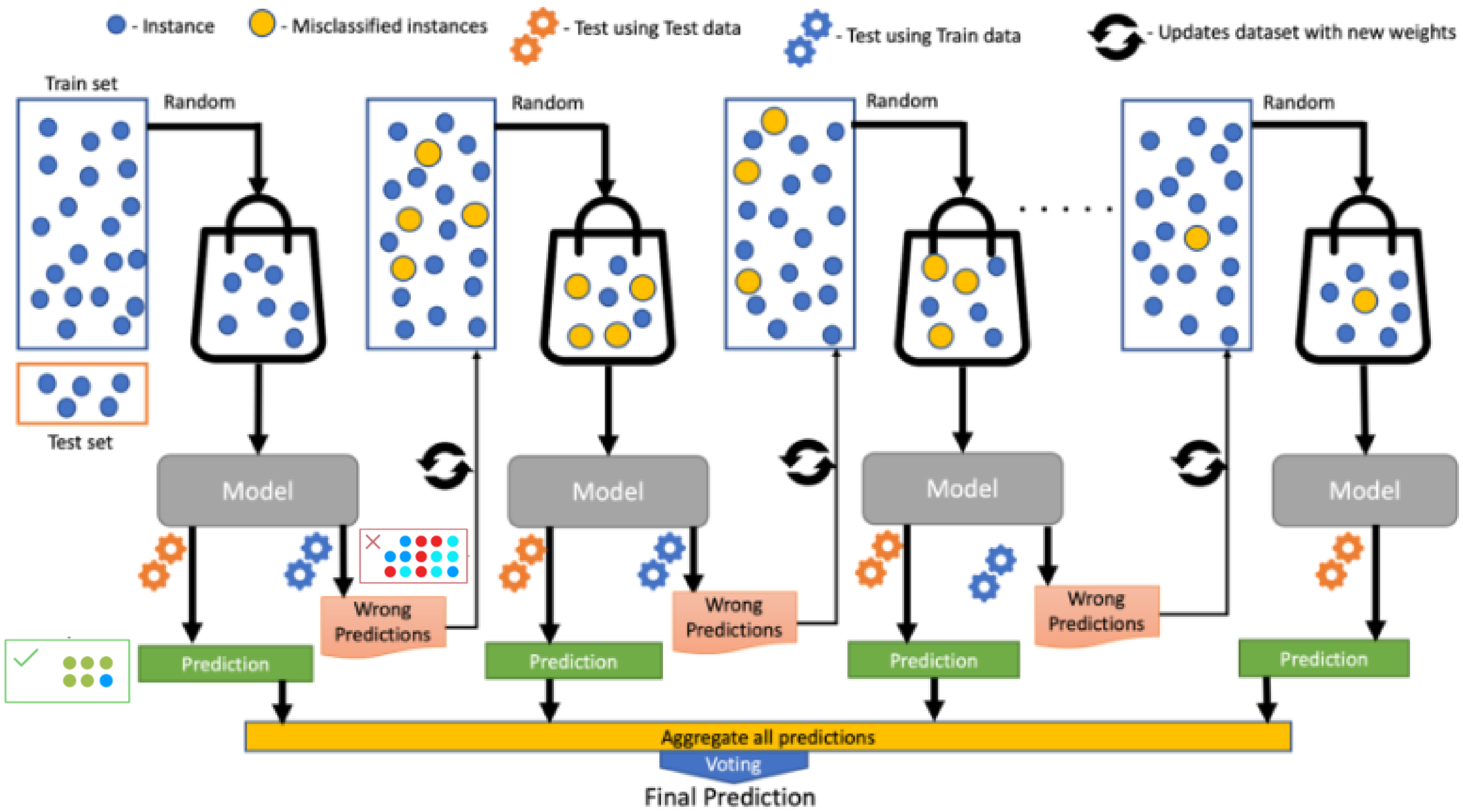
Build and train regressor model to using SHAP algorithm, explain the impact of features in predictions

Methodology



Models and Experiments with X-AI

XGBoost Regressor Architecture



Bootstrapping

Bagging

Training

Aggregating

XGBoost train and hyperparameter tuning

```python
import xgboost as xgb
from sklearn.model_selection import GridSearchCV

# define grid hyperparameters
xgb_params = {
    "n_estimators": range(100, 500, 100),
    "learning_rate": [1e-3, 0.01, 0.05, 0.1],
    "max_depth": range(8, 32, 8),
    "subsample": [0.8, 0.9],
    "colsample_bytree": [0.8, 0.9]
}

# define XGBoost regressor with hyperparameter tuning
xgb_grid = GridSearchCV(estimator = xgb.XGBRegressor(),
                        param_grid = xgb_params,
                        cv = 5,
                        scoring = "r2",
                        verbose = True,
                        n_jobs = -1
                    )
# fit the model
xgb_grid.fit(x_train, y_train)
```

Best model

Parameters	Best values
n_estimators	400
learning_rate	0.1
max_depth	8
subsample	0.8
colsample_bytree	0.8

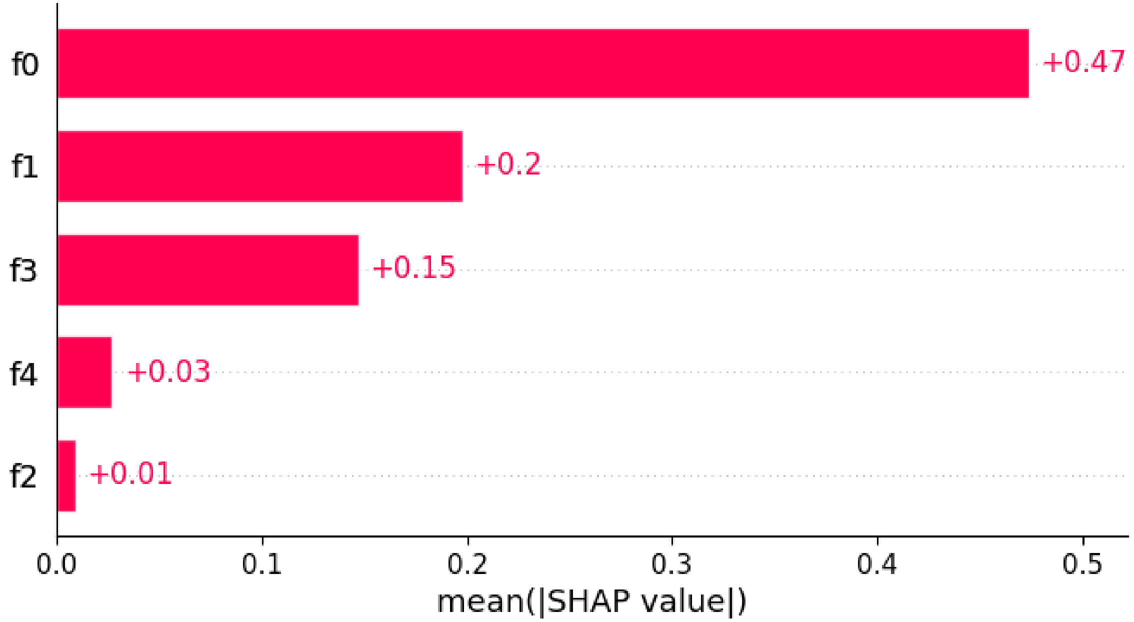
Disk: 1290 KB

Model Inference

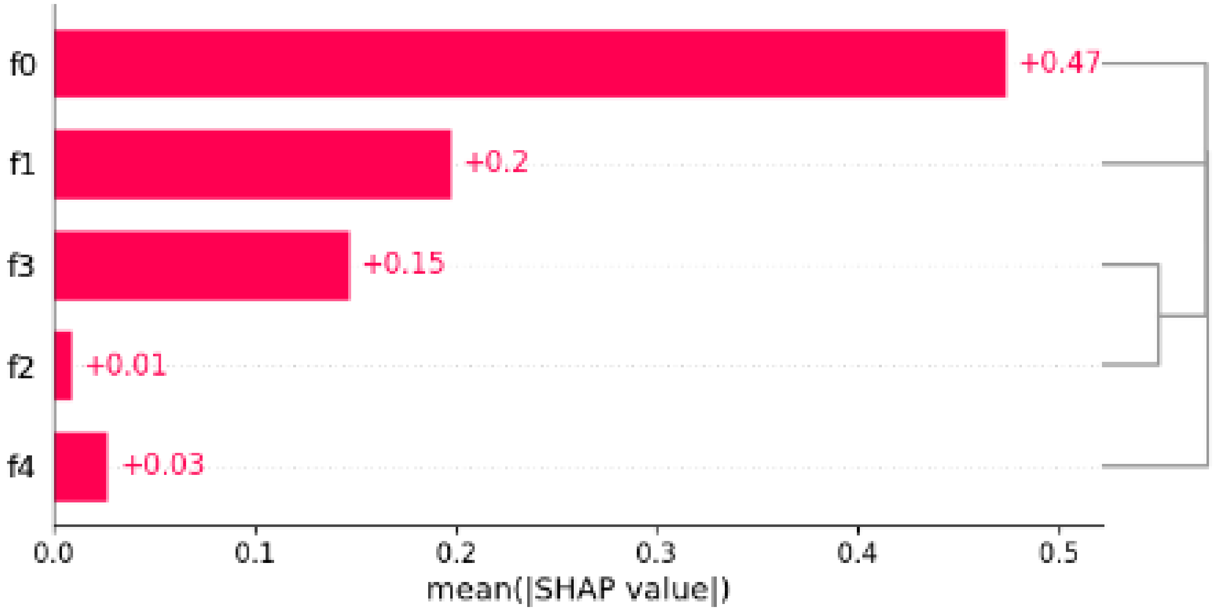
Train R^2 score	Test R^2 score
99.93%	99.95%

XGBoost with SHAP for Global explanations

Feature importance bar plots

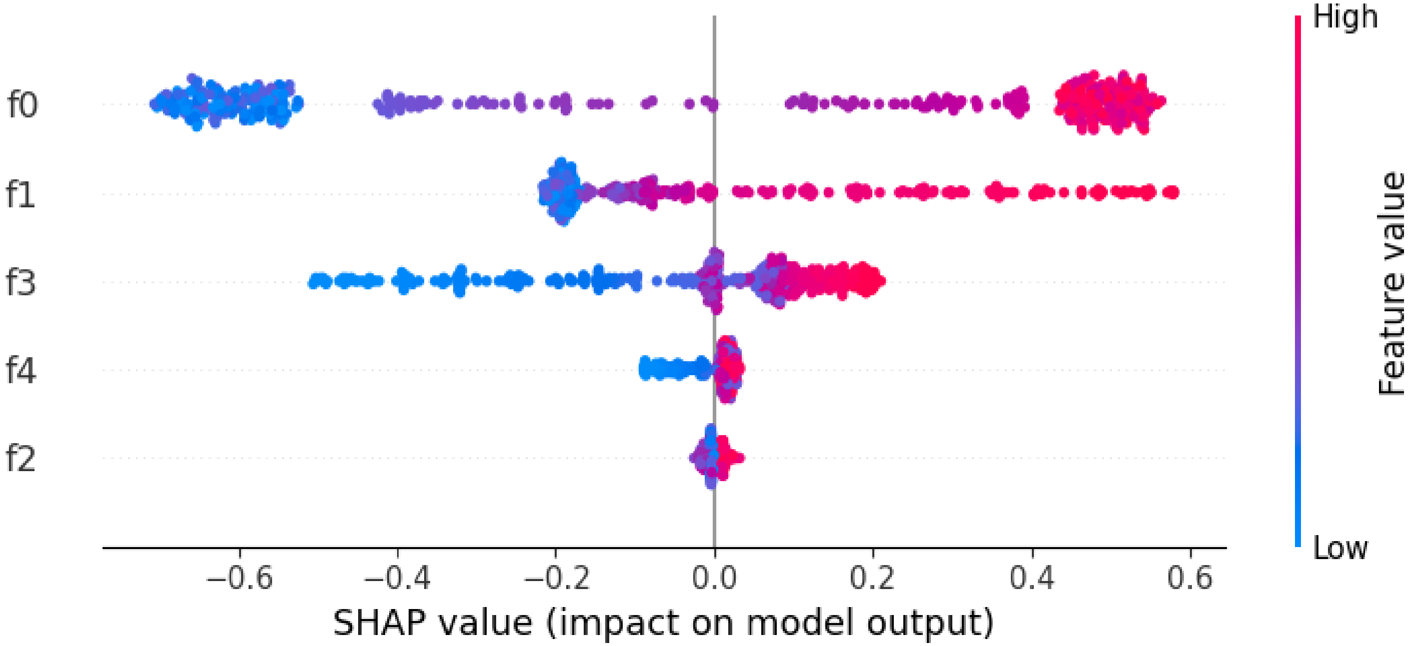


Hierarchical clustering among features

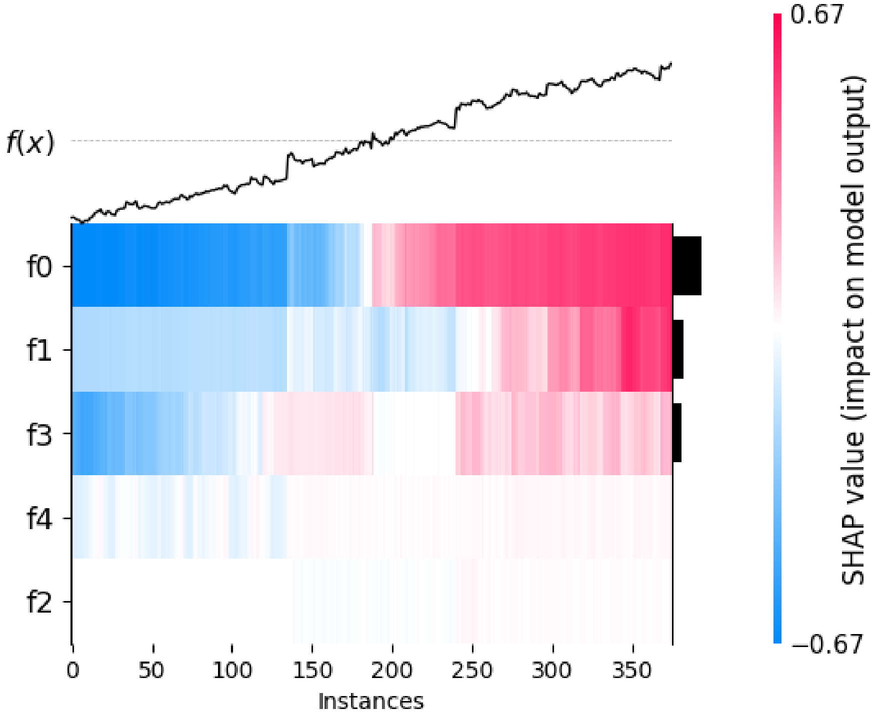


XGBoost with SHAP for Global explanations

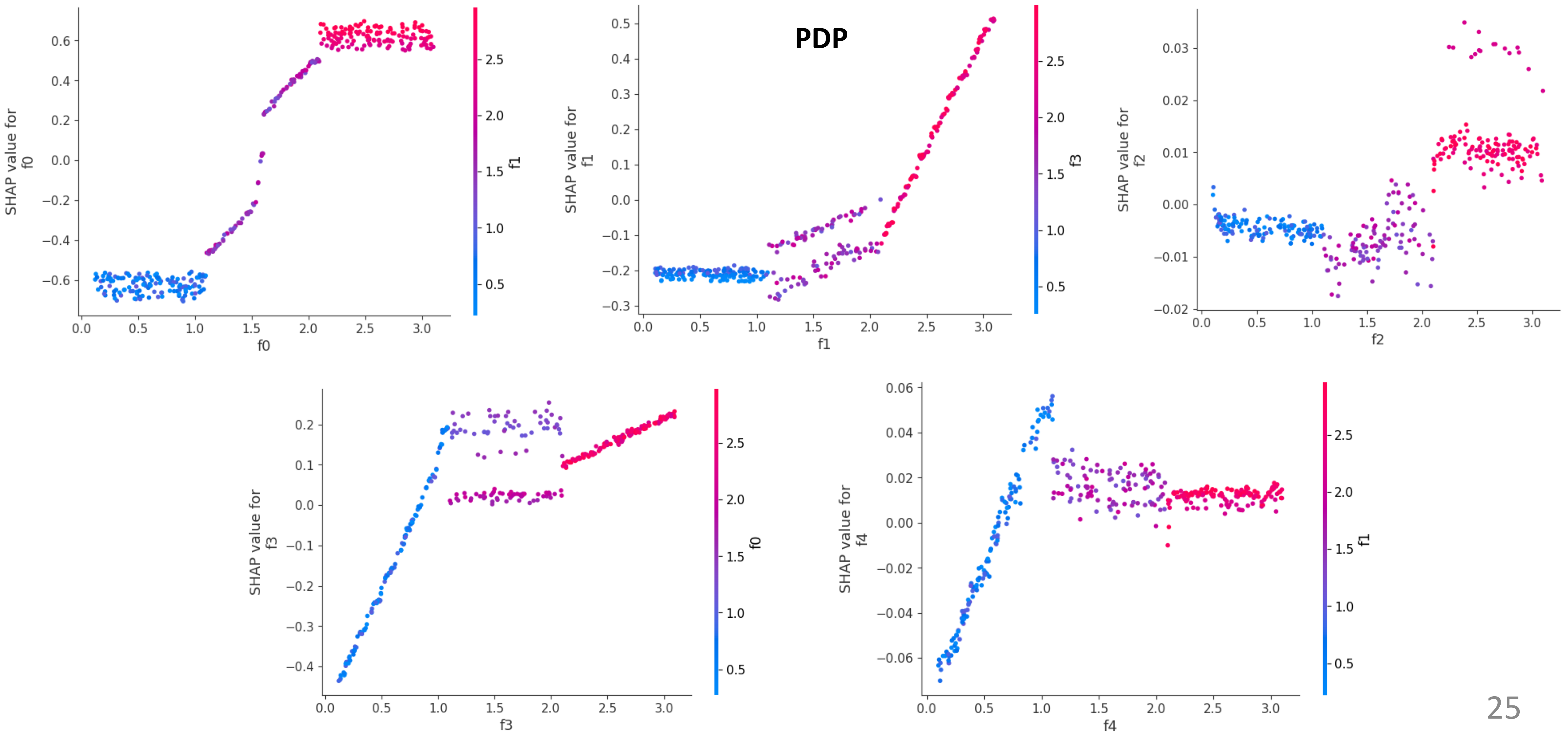
Summary plots



Heat maps

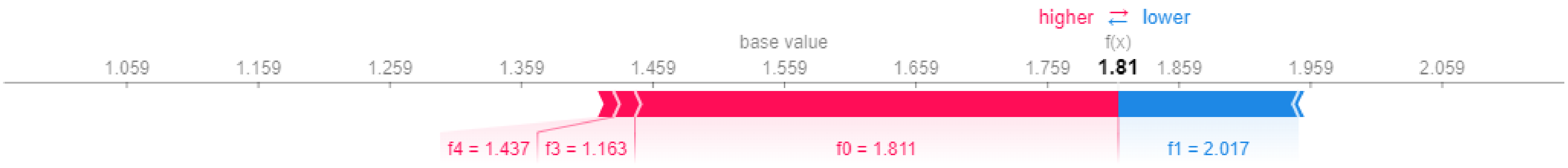


XGBoost with SHAP for Global explanations

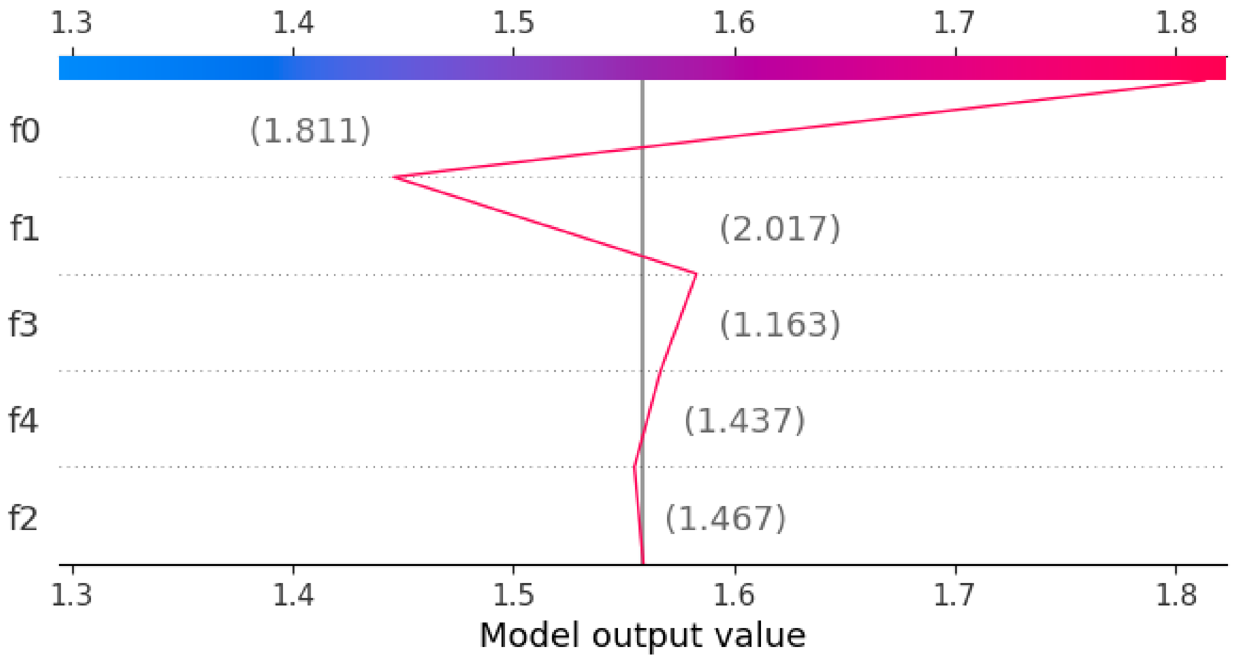


XGBoost with SHAP for Local explanations

Force plot

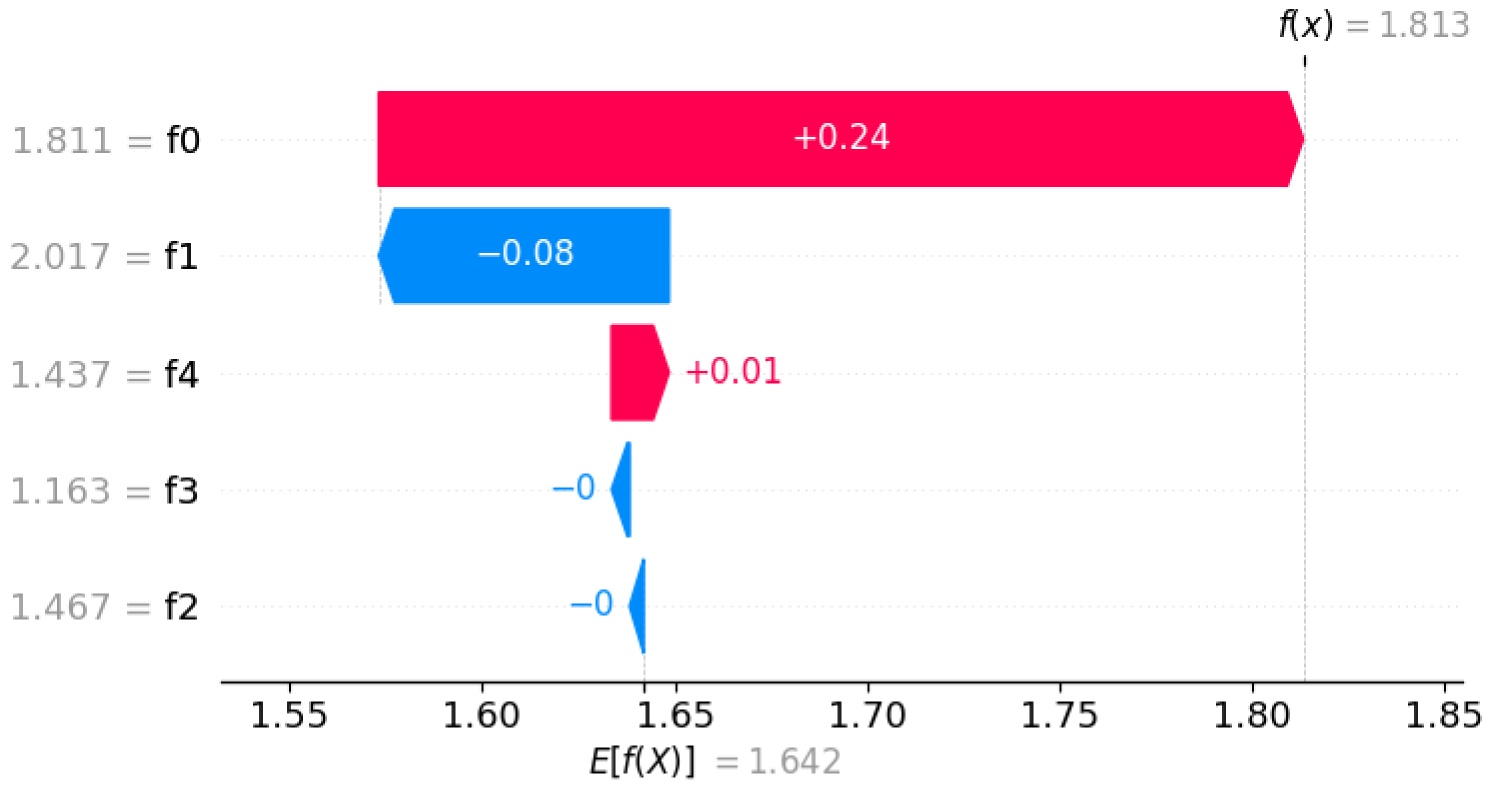


Decision plot



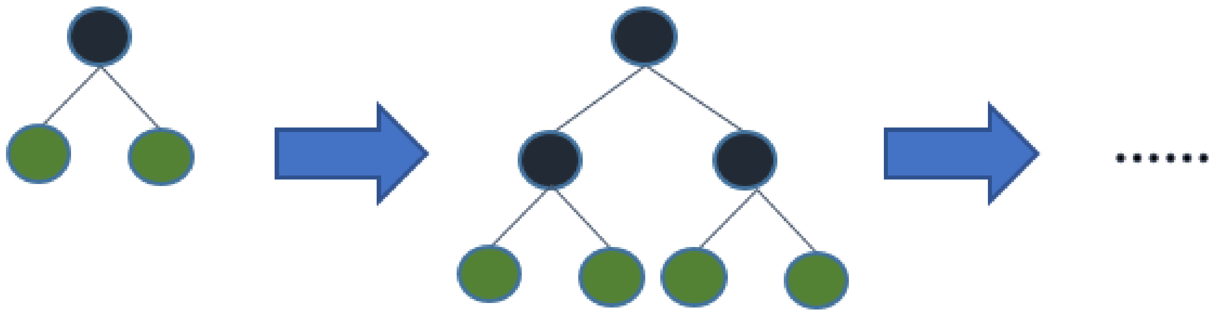
XGBoost with SHAP for Local explanations

Waterfall plots



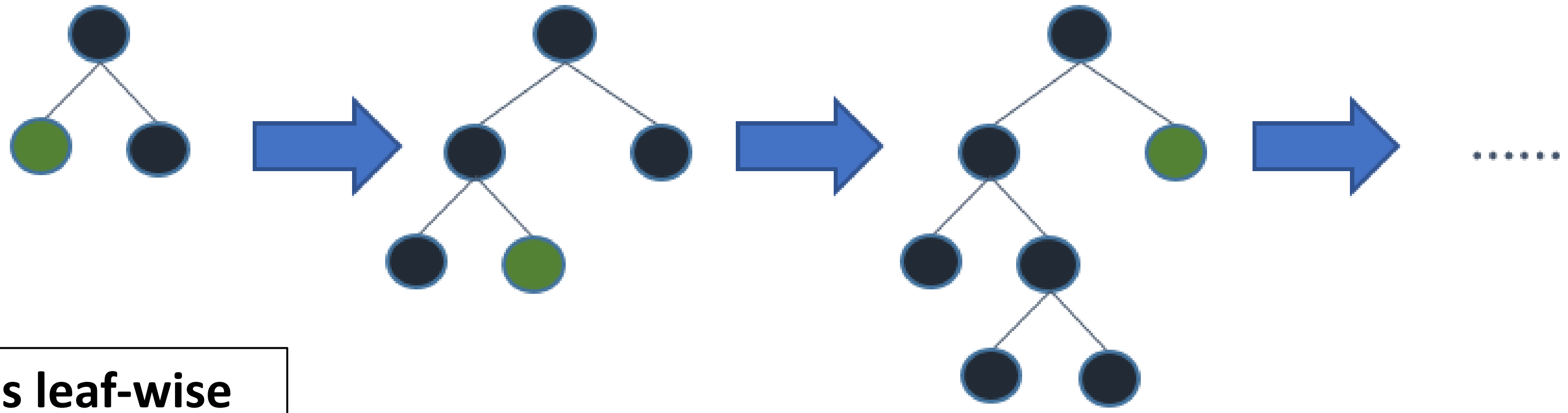
LightGBM Regressor Architecture

Grow trees by level (deep)-wise



Level-wise tree growth

Grow trees by leaf-wise (best-first)



Leaf-wise tree growth

LightGBM grows trees leaf-wise instead of level-wise!

LightGBM train and hyperparameter tuning

```python
import lightgbm as lgbm
from sklearn.model_selection import GridSearchCV

# define grid hyperparameters
lgbm_params = {
    "max_depth": range(8, 32, 8),
    "num_leaves": range(30, 80, 10),
    "learning_rate": [1e-3, 0.01, 0.05, 0.1],
    "bagging_fraction": [0.7, 0.8, 0.9]
}

# define the LightGBM regressor
lgbm_grid = GridSearchCV(estimator = lgbm.LGBMRegressor(),
                         param_grid = lgbm_params,
                         cv = 5,
                         scoring = "r2",
                         verbose = False,
                         n_jobs = -1
                     )
# fit the model
lgbm_grid.fit(x_train, y_train)
```

Best model

Parameters	Best values
max_depth	24
num_leaves	30
learning_rate	0.1
bagging_fraction	0.7

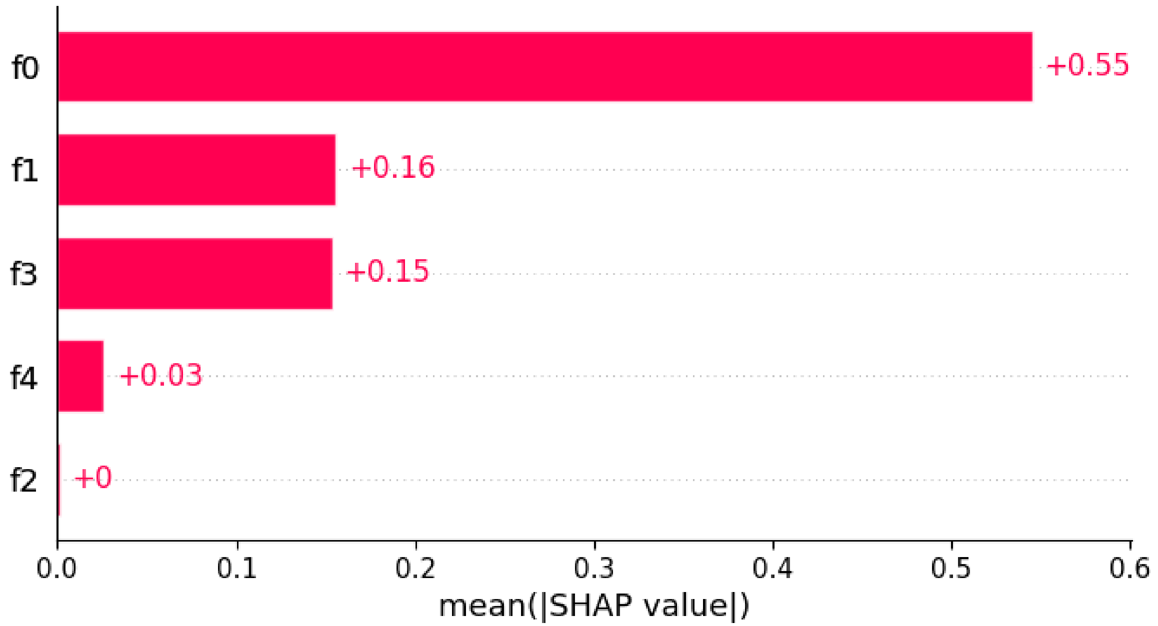
Disk: 279 KB

Model Inference

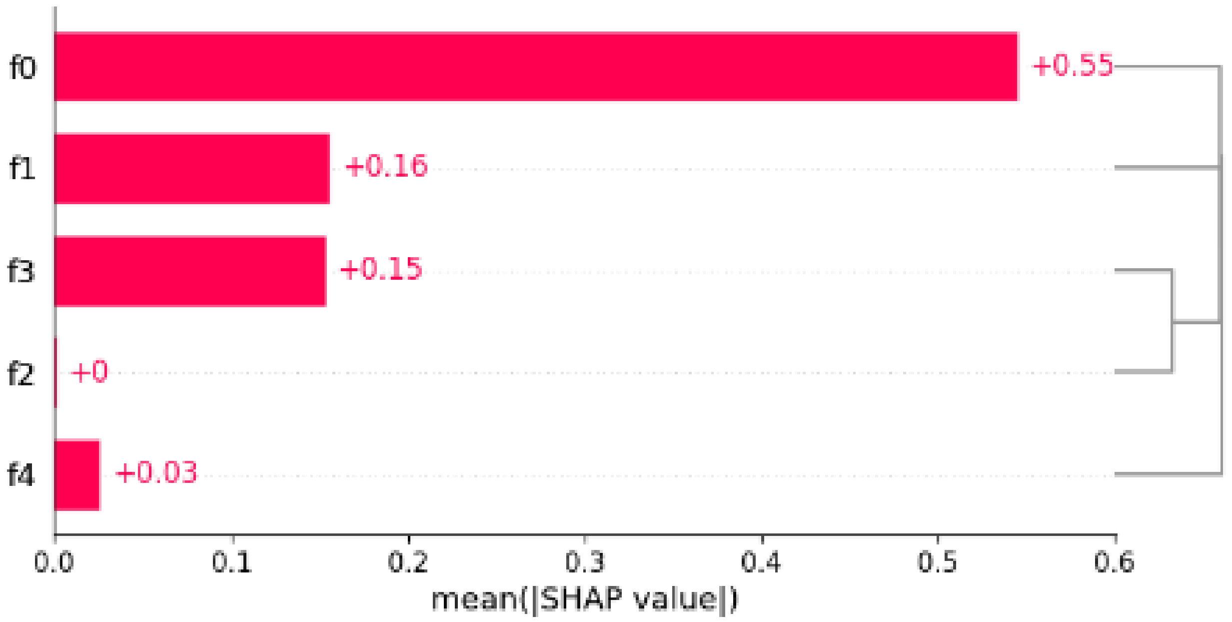
Train R^2 score	Test R^2 score
99.96%	99.97%

LightGBM with SHAP for Global explanations

Feature importance bar plots

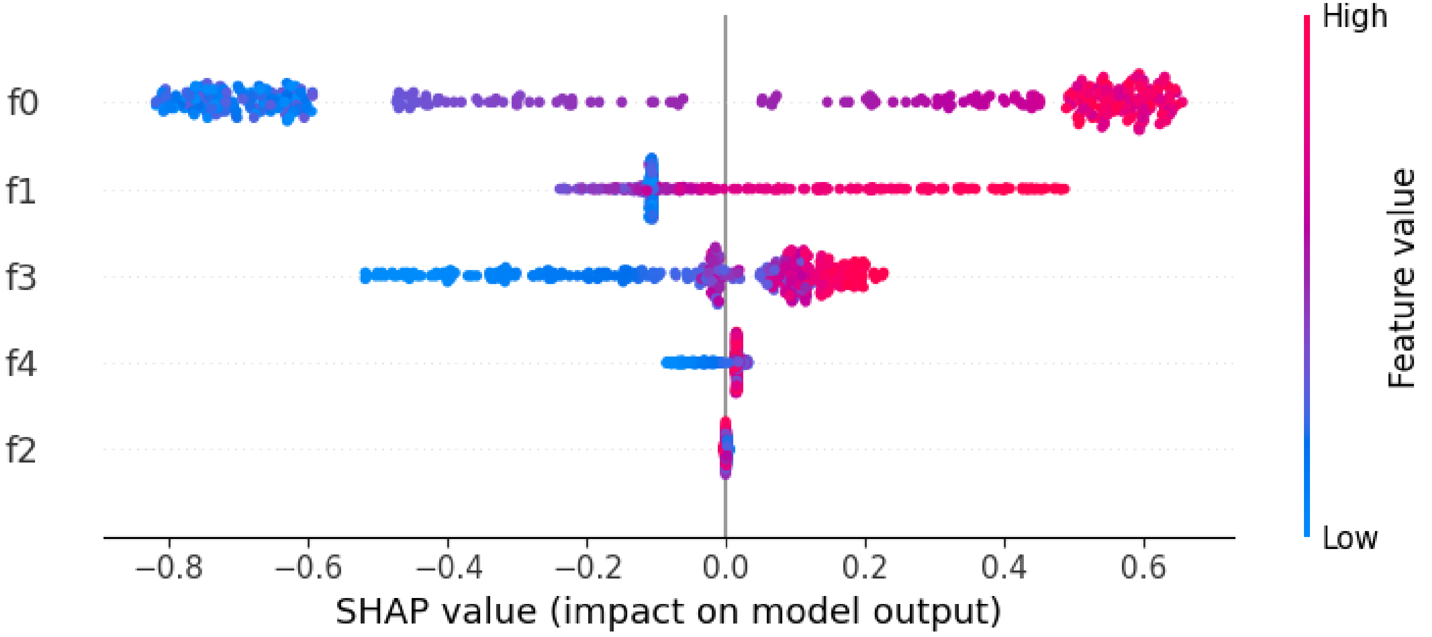


Hierarchical clustering among features

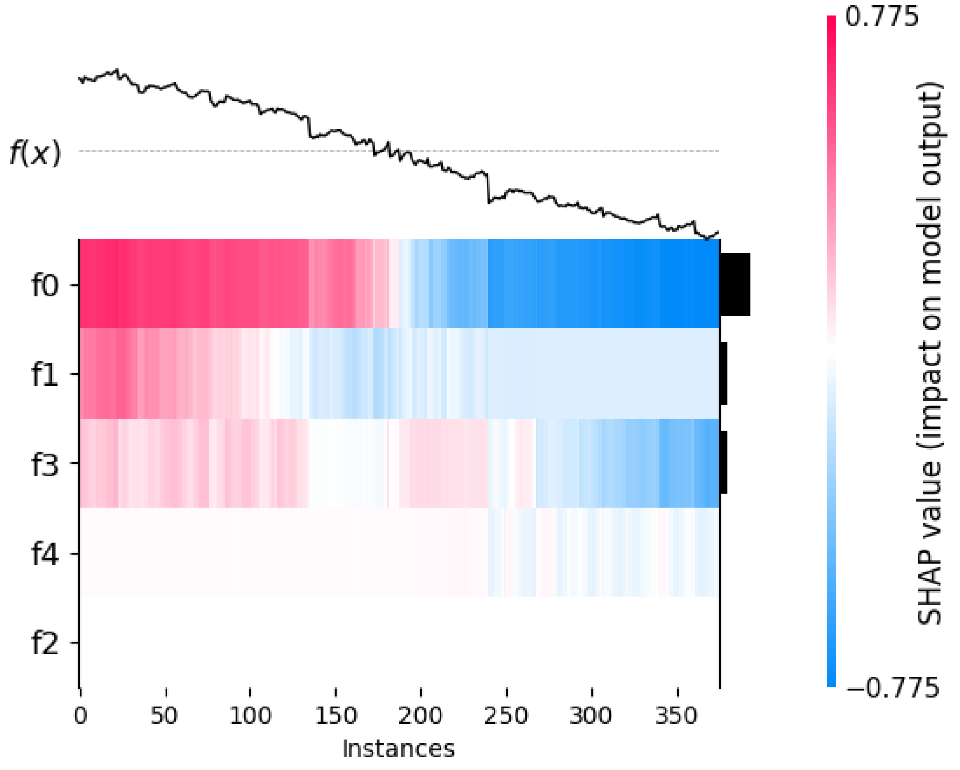


LightGBM with SHAP for Global explanations

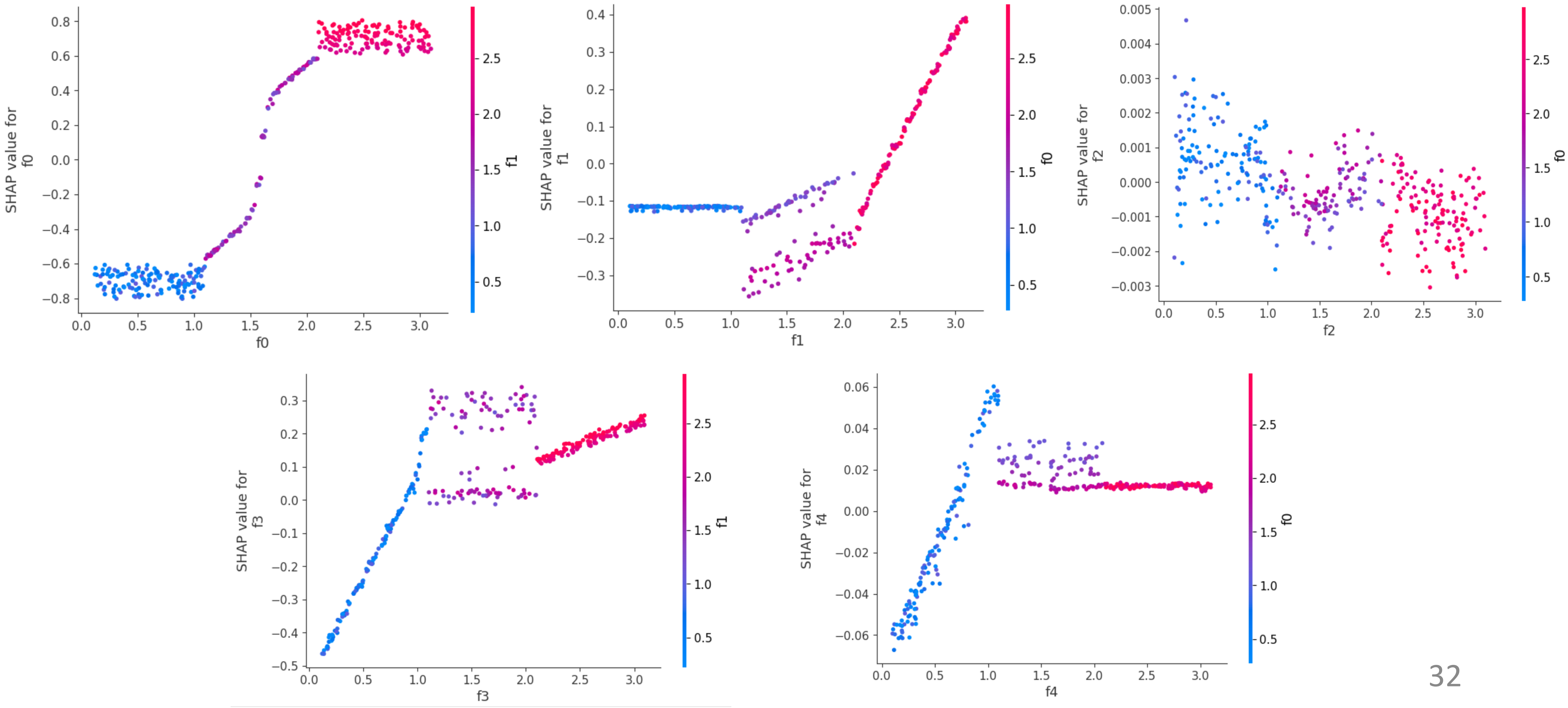
Summary plots



Heat maps

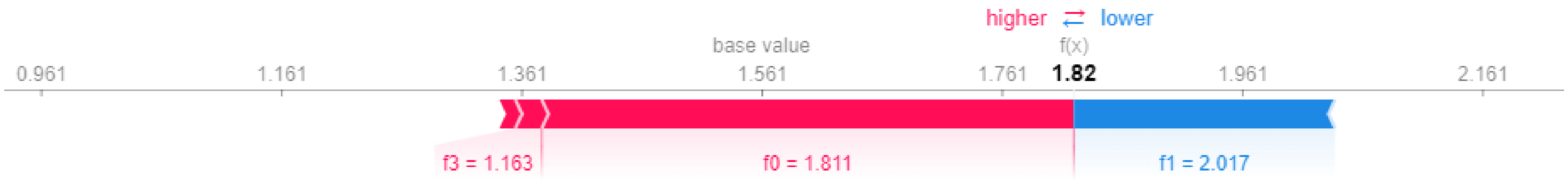


LightGBM with SHAP for Global explanations

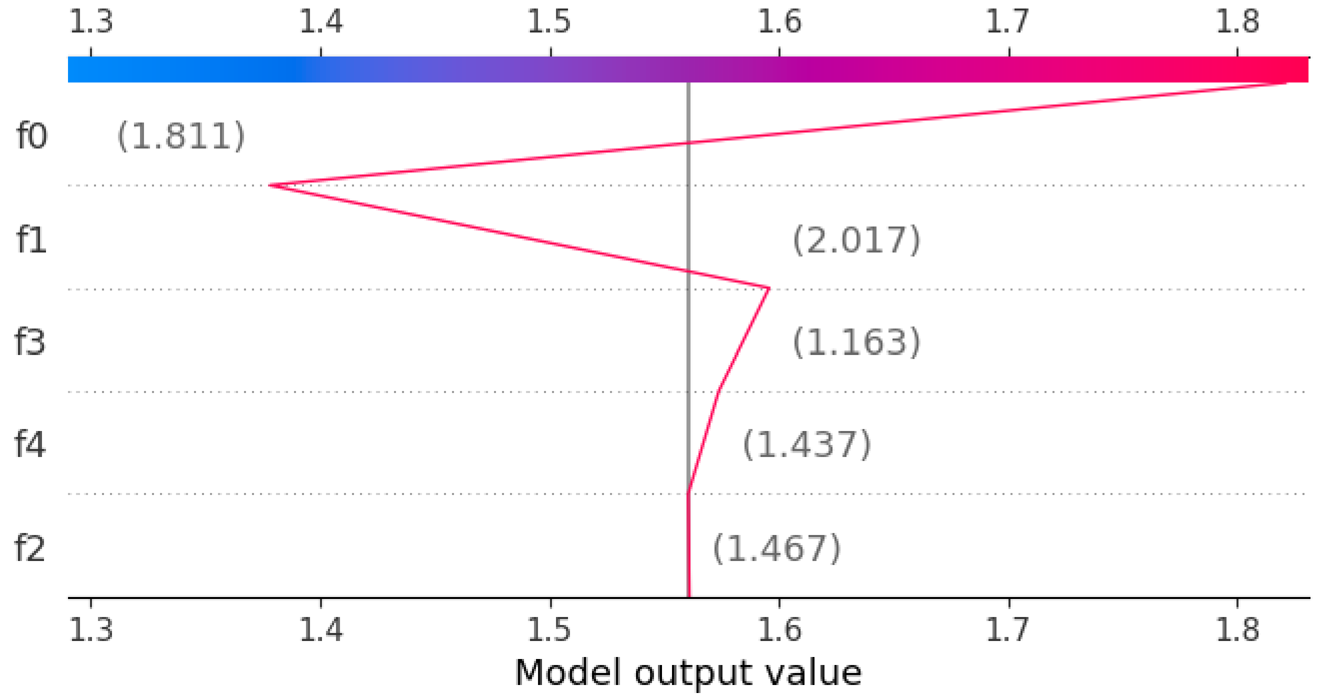


LightGBM with SHAP for Local explanations

Force plot

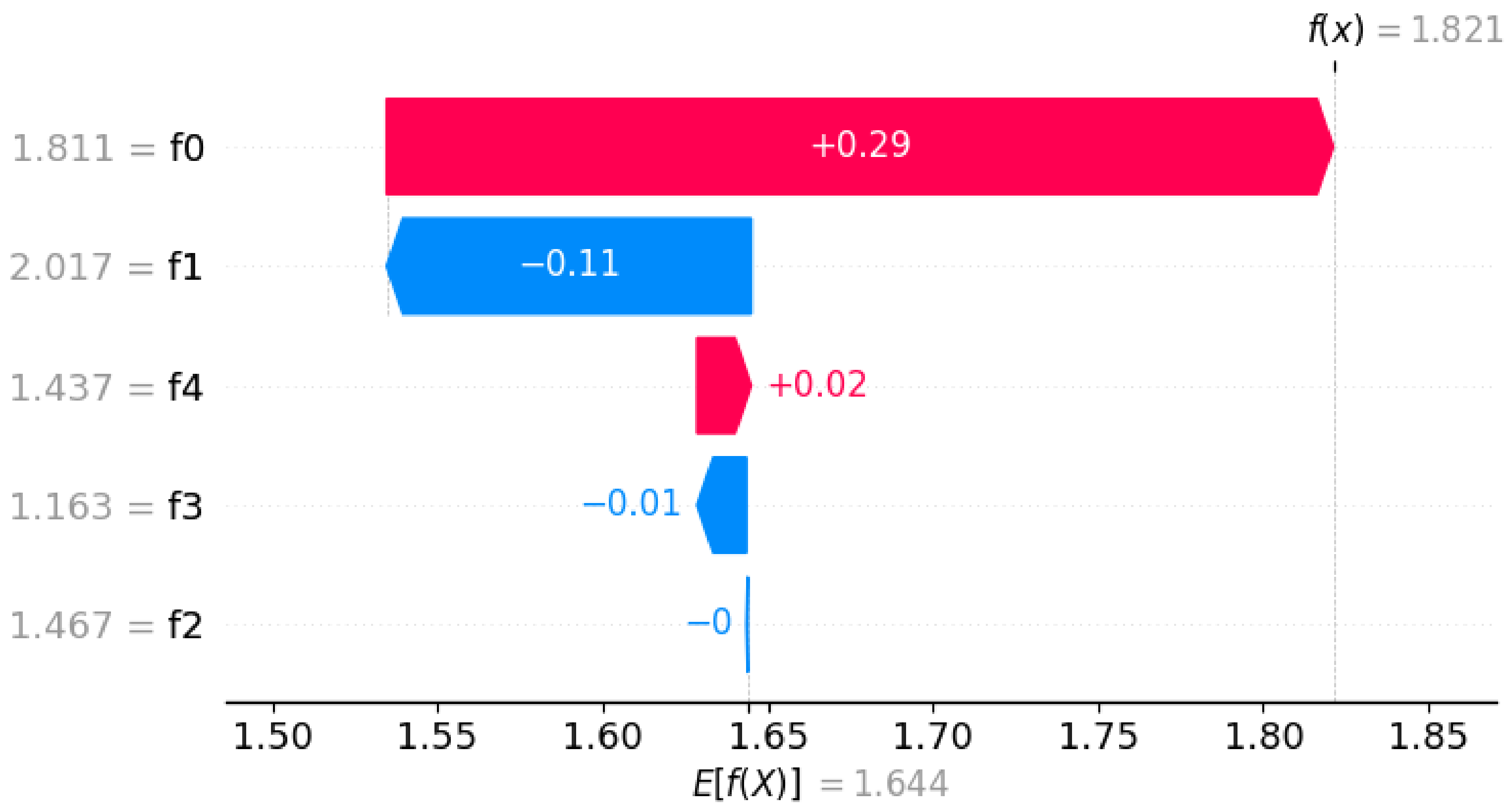


Decision plot



LightGBM with SHAP for Local explanations

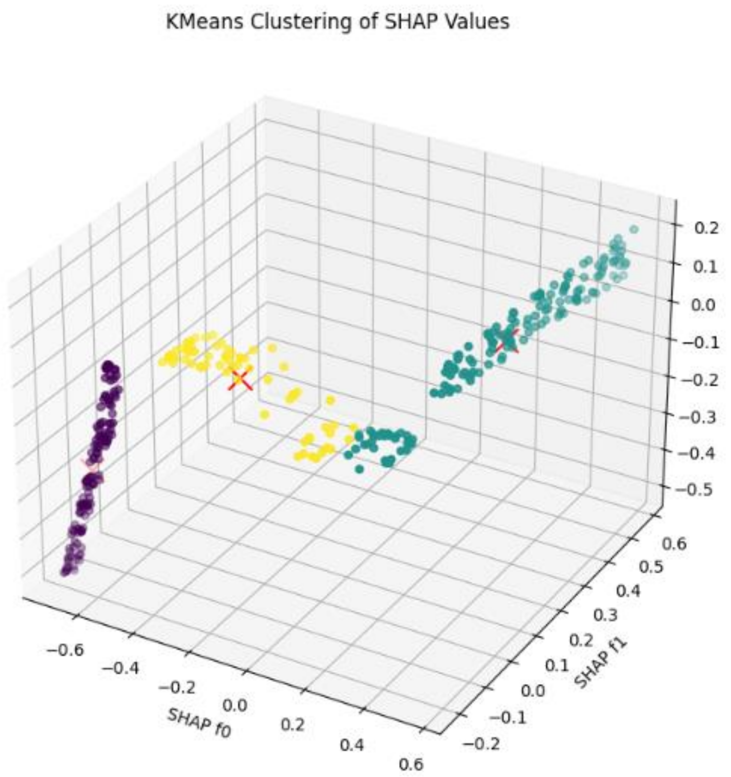
Waterfall plots



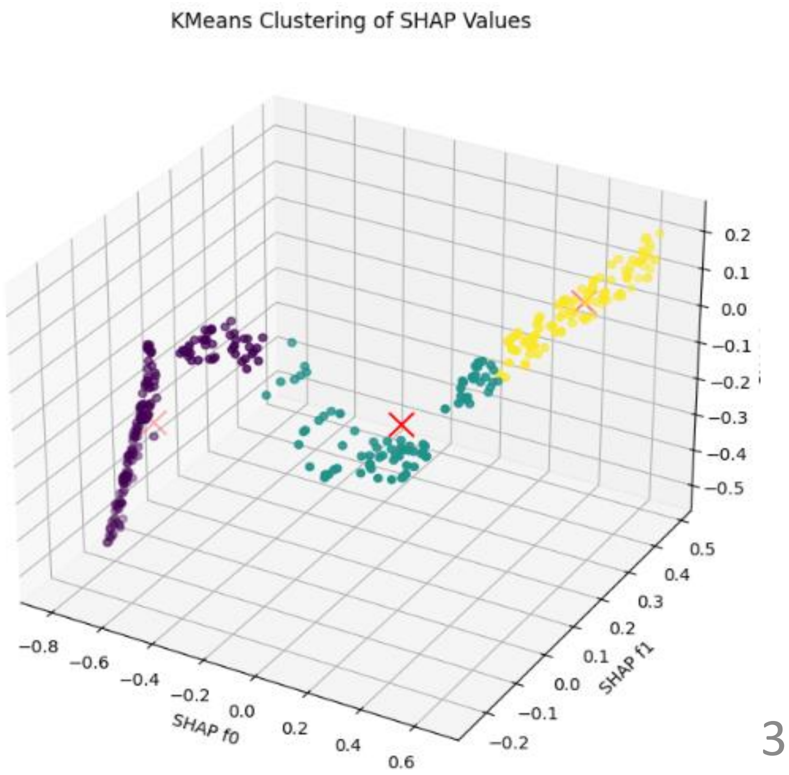
Clusterization of SHAP values with K-means

Number of clusters = 3; Dimensionality reduction to 3D: considering features **f0**, **f1** and **f3**

Clusterization for XGBoost SHAP values



Clusterization for LightGBM SHAP values



References

- [1] Masís, S. (2021). Interpretable Machine Learning with Python: Learn to build interpretable high-performance models with hands-on real-world examples. Packt Publishing Ltd.
- [2] Bhattacharya, A. (2022). Applied Machine Learning Explainability Techniques: Make ML models explainable and trustworthy for practical applications using LIME, SHAP, and more. Packt Publishing Ltd.
- [3] Thampi, A. (2022). Interpretable AI: Building explainable machine learning systems. Simon and Schuster.
- [4] Munn, M., & Pitman, D. (2022). Explainable AI for Practitioners. " O'Reilly Media, Inc.".
- [5] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

Thank you for your attention!

Эспинола Ривера, Хольгер Элиас
espinolarivera.h@edu.spbstu.ru
holtech94@gmail.ru