Fast and Accurate Feature Impact Curves for ML-powered dynamic dashboards

2nd CEU Vienna Data Analytics Jamboree

Márton Nagy | CEU Gabors Data and AI Lab

Vienna, 5 December 2025

Contents
of today's talk

1. Why should you care? - a brief intro

2. The problem
2.1. Context: a fictional case
2.2. The current approach and its shortcomings

3. Technical background
3.1. Overview of global, model-agnostic tools
3.2. SHAP dependency plots
3.3. Partial Dependence Plots
3.4. Accumulated Local Effects

4. The solution
4.1. The analytically correct option: ALE with some modifications
4.2. The pragmatic solution: rebasing SHAP values

1. Why should you care?

2. The problem

2.1. Context: a fictional case

If we let users dissect our models on a dashboard without expert oversight, we have to make sure that results stay accurate even in weird cases.

Risk prediction problem

Predict the **probability of individuals doing a certain thing in a given timeframe**, e.g. customers leaving a telco provider in the next three months.

Results presented on a dynamic dashboard

Users can **slice and dice the data** and predictions to look at **results that are relevant** for them.

One of the key tools: impact curves

Users would like to see a **simple line chart** of how the **values of a feature** are associated with the **predicted risk**.

2.2. Naive approach and its shortcomings

01 Using SHAP for impact curves

The current approach is a **SHAP dependency plot**: the expected mean of SHAP values conditional on the feature's values.

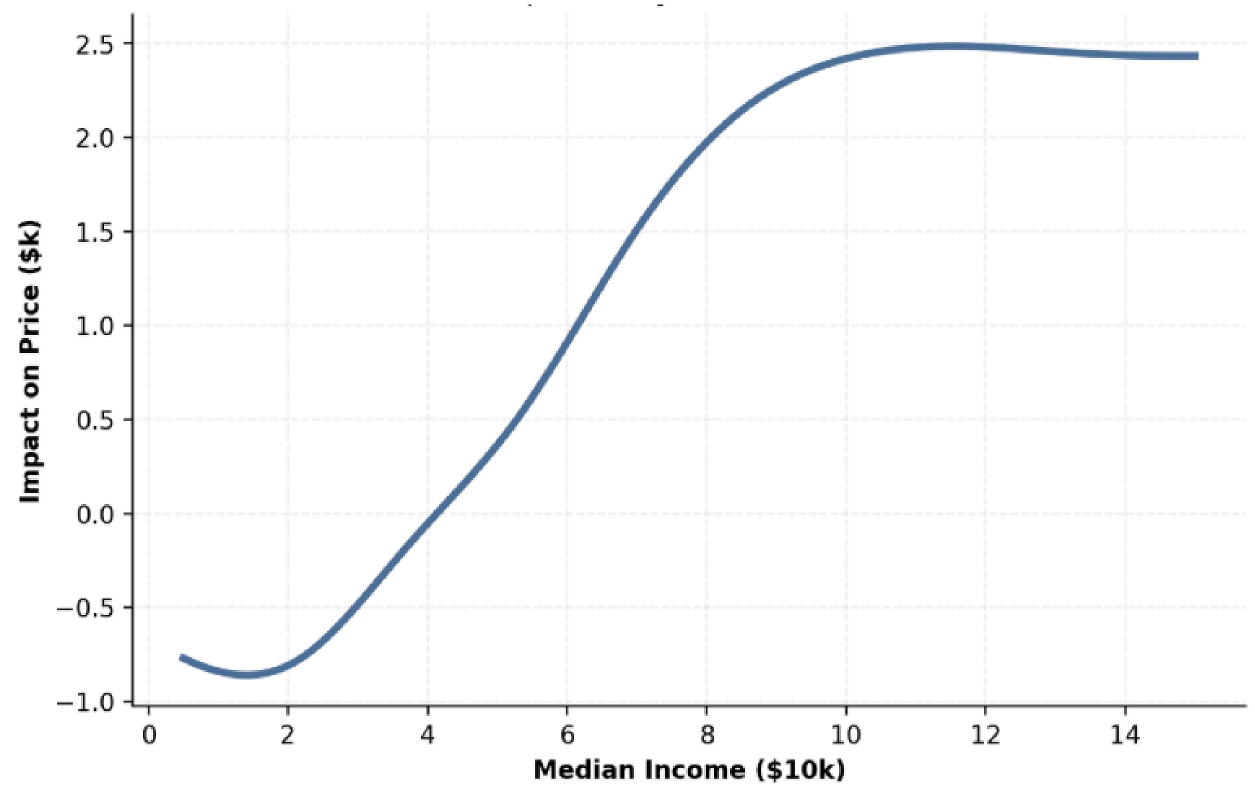
02 It has some very nice properties...

It is easily **interpretable** by non-technical users. SHAP values can be **pre-calculated** so that the plots appear practically **instantenously** on the dashboard.

03 ...but some important caveats as well

Global SHAP values are **always relative to the global baseline** and may **hide** important **local nuances when filters are applied**: an interpretation and a specificity issue.

An example SHAP DP on the California Housing dataset

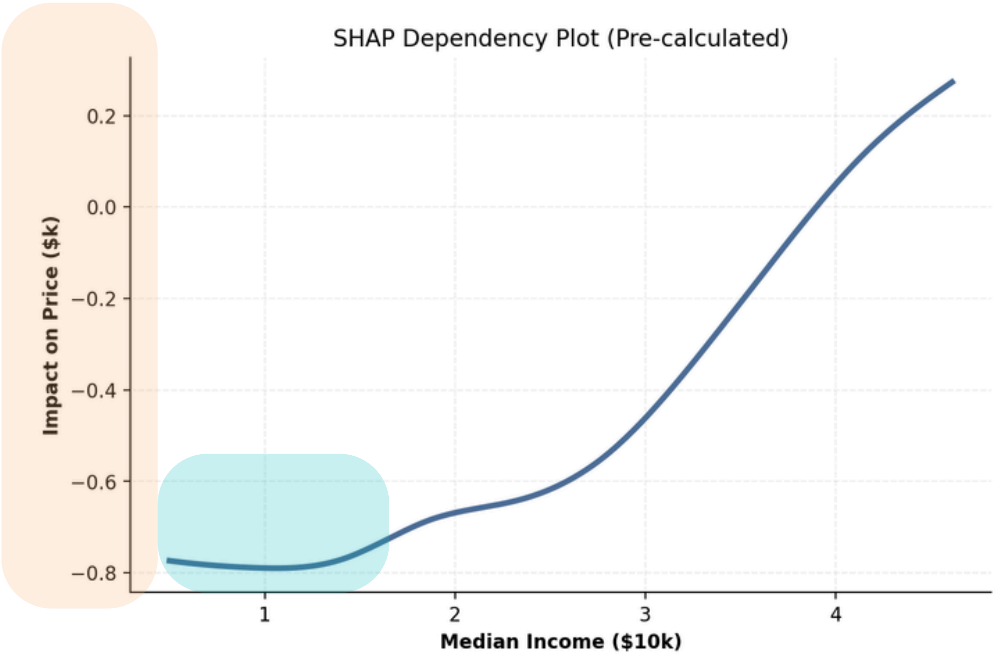


3. Technical background

3.1. Overview of model-agnostic feature impact xAI tools

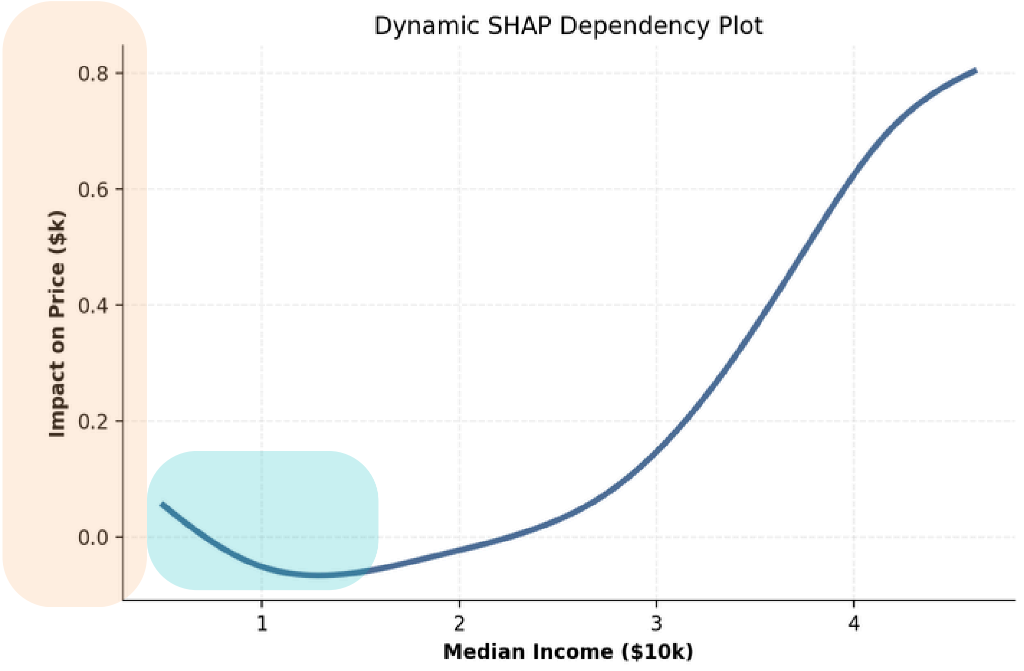
	SHAP Dependency Plots	Partial Dependence Plots	Accumulated Local Effects
PRO	• Can be **precomputed** • No hyperparameters • Inherently **deals with correlated features** • Scatterplot can show heterogeneity	• Remains interpretable and **specific when filters** are applied • Almost no hyperparameters (except grid size) • ICE curves can show heterogeneity	• Remains interpretable and **specific when filters** are applied • **Works** even if features are **correlated** • Can be **pre-computed**(ish)
CON	• Global values are **off when filters are applied**: both shape and scale can be mismatched • **Dynamic** re-calculation is computationally **expensive**	• **Cannot** be **precomputed** • Assumes **independence** of features • Gives weight to **unrealistic** parts of the feature space	• **Binning method** heavily **influences** the results • Traditional approach results in **rugged plots** • Cannot show heterogeneity

3.2. SHAP Dependency Plots



SHAP DP on a filtered sample: global values filtered...

...vs. re-computed SHAP values on the filtered sample



01 Basic interpretation

Average expected **contribution** of the feature to the prediction (both **main and interactions**) **conditional** on the **feature value**, **relative** to the global **average prediction**.

02 Strengths

- Principled and **easy to understand.**
- Global values **can be pre-computed** so that only the plotting happens at runtime.

03 Weaknesses

- Pre-computed values are always relative to the global baseline, not the filtered one → **an interpretation issue**
- SHAP values depend on the underlying joint distribution of the features: if filtering changes this, there can be changes in the curve shapes → **a specificity issue**
- Re-computing SHAP values would kill the UX → **a speed issue**

3.3. Partial Dependence Plots

01 Basic interpretation

The **average predicted output** of the model **when** the **feature** is set to **a specific value for every observation.**

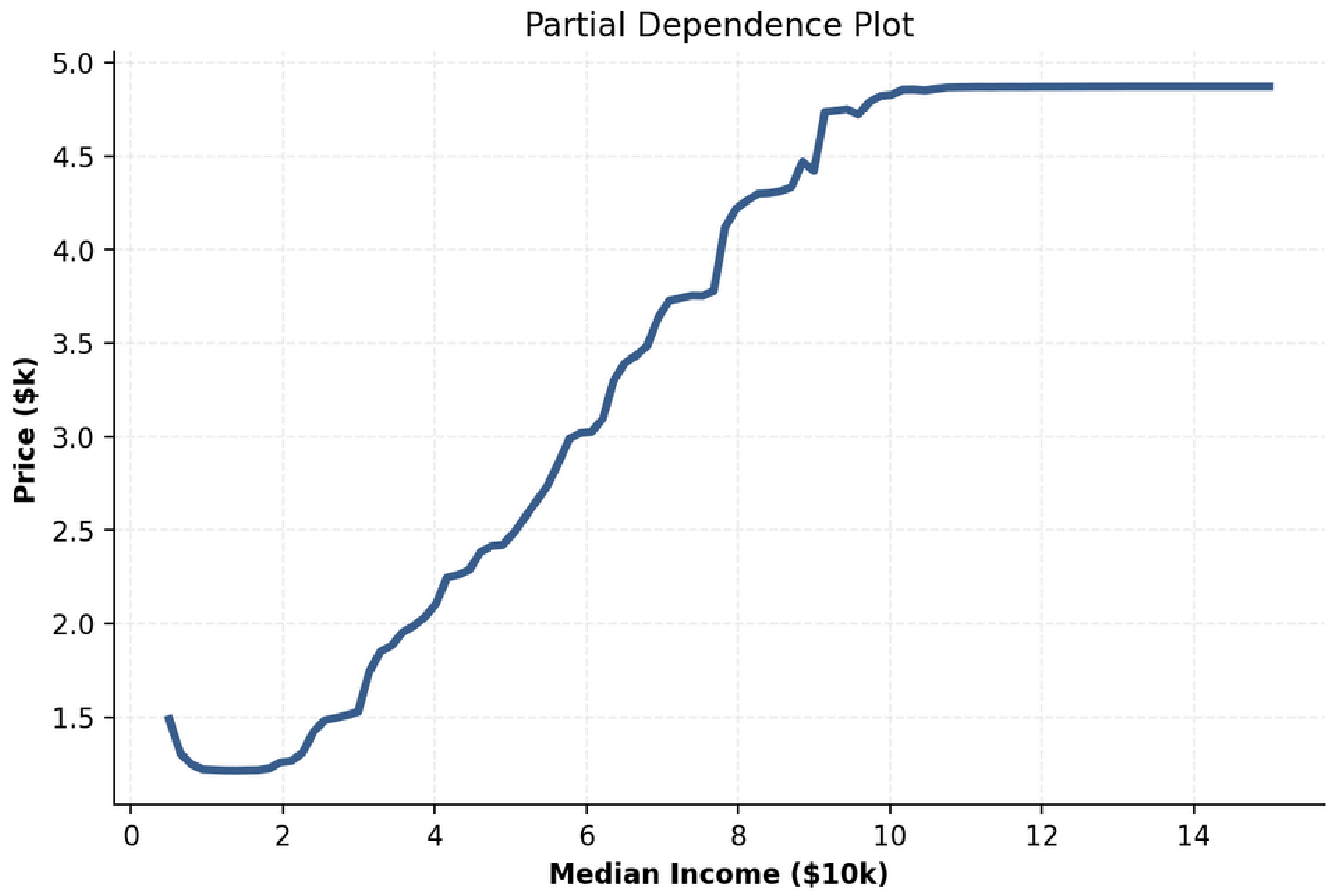
02 Strengths

- **Accurate even after filters**
- **Easiest to understand** both the calculation and the interpretation.

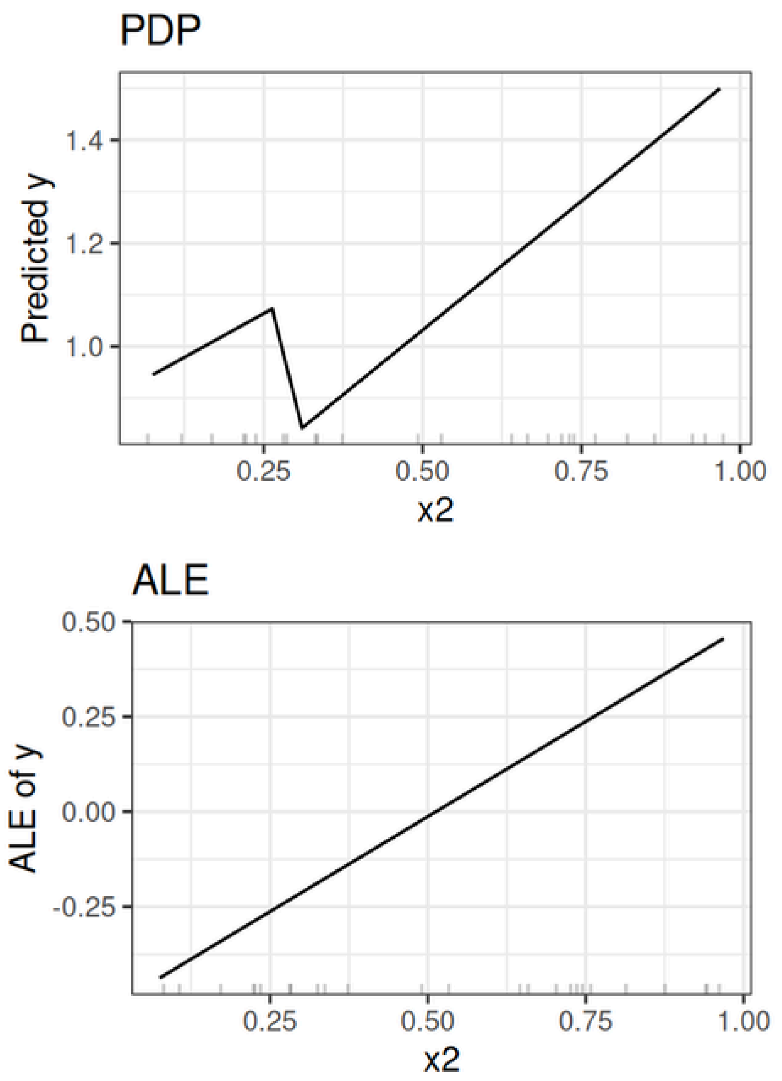
03 Weaknesses

- **Cannot be pre-computed**, computation includes N times the grid size model prediction calls
- **Weighs in unrealistic scenarios** (e.g. person with 2m height and 40kg weight)
- **Unreliable when features are correlated** (as in reality we cannot change one while keeping the other fixed).

An example PDP on the California Housing dataset



3.4. Accumulated Local Effects Curves



PDP vs ALE plots on a synthetic example where two predictors are heavily correlated

DGP is simply linear, but PDP fails to capture this, as it weighs such parts of the feature space where no observation can lie.

Source: Interpretable Machine Learning by Christoph Molnar, Fig. 20.5

01 Basic interpretation

Average change in prediction when the **feature is varied within small intervals**, **accumulated** over the feature range, **relative to** the sample's **average prediction**.

02 Strengths

- In **most cases** shows the **same** shape **as PDP**, only the baseline is different
- **Robust to correlated** features
- **Robust to filtering**

03 Weaknesses

- The standard version **cannot be pre-computed**, and computation takes time (though less than PDP)
- Relies on binning the feature range, and the **binning choice** can heavily **influence the results**
- Standard ALE plots can look **rugged**, **hindering across-bin interpretation**

4. The solution

4.1. Analytically correct: ALE with modifications

Problems faced

Standard version cannot be pre-computed

Binning is often arbitrary

Plots are rugged, making cross-bin interpretation hard

Solution

Implement ALE as per the mathematical definition

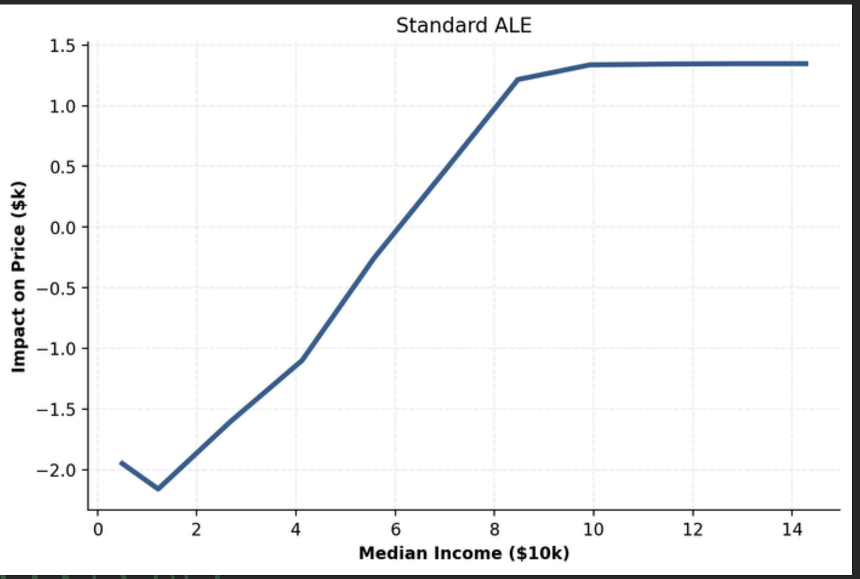
ALE is defined as the **centered integral of the partial derivative** function w.r.t the feature. The **partial derivative can be pre-computed** for every observation and feature (resulting in the **Jacobian matrix)**. For tree-based models, we need to smooth a bit, but the idea is the same.

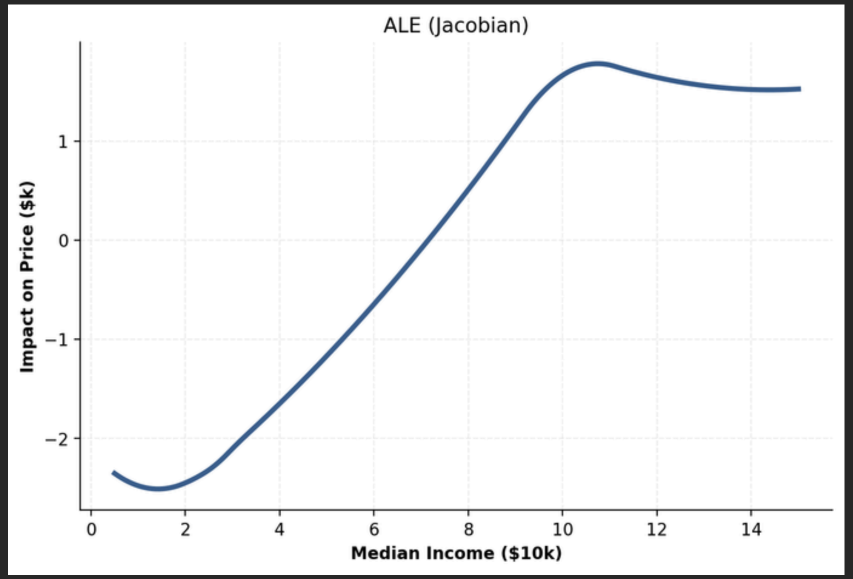
Recent literature implements dynamic programming based binning

The idea is to **minimize the within-bin heterogeneity** of local effects to get reliable curves. Implemented by Gkolemis (2023). **Algo is fast** even for large dataset with pre-computed Jacobian.

Fit a PLS on the Jacobian → integrated curve is smooth

Accumulating the within-bin average partial derivative over the bins would result in the standard ALE plot. However, if we **fit a linear spline model with bin edges as knots**, the **integrated** function is built up of **quadratic segments**, resulting in a smoother and more interpretable curve.





4.2. Pragmatic: Rebase global SHAP DP

Problems faced

The scale of filtered global SHAP DPs is off

The shape of filtered global SHAP DPs is off

Solution

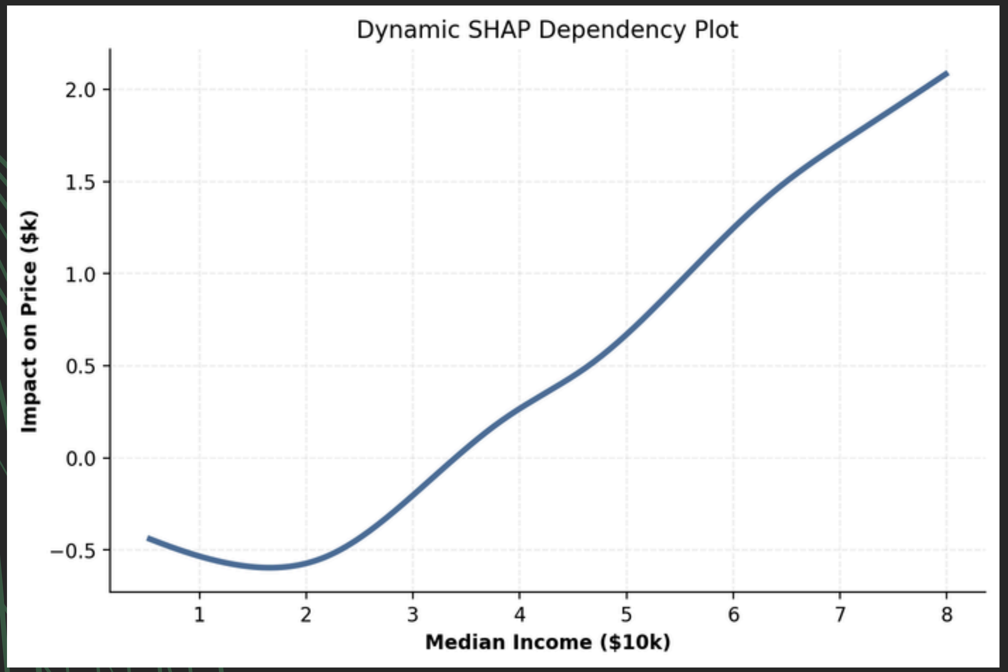
Just a re-basing issue with simple workaround

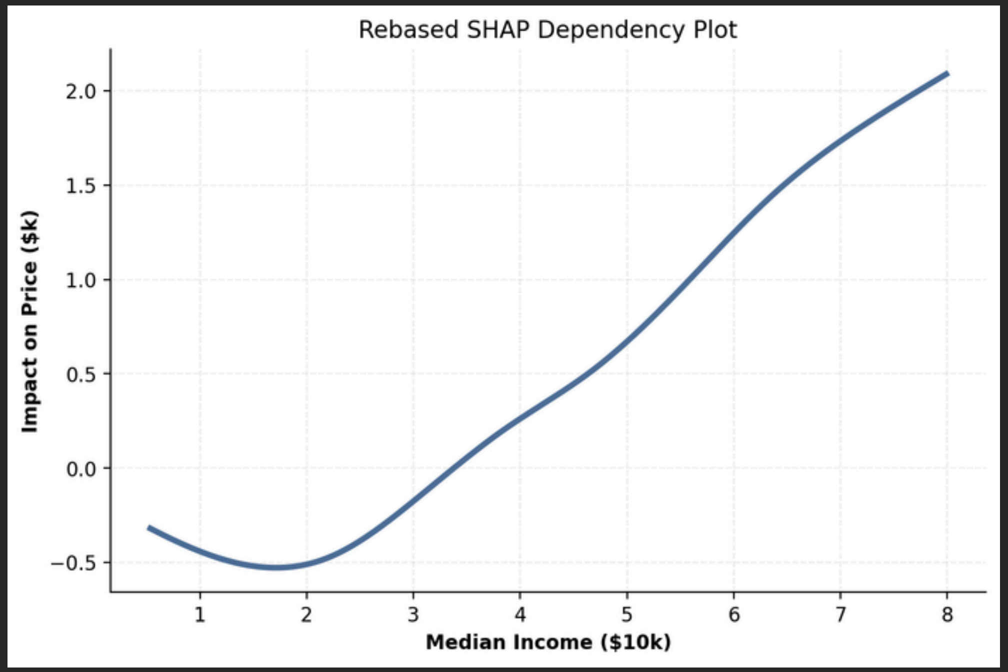
SHAP values should always be relative to the mean prediction of the sample used as the background dataset. Thus **we can easily re-center the filtered global SHAP DP** to be relative to the filtered sample's average prediction.

Cannot be solved without re-computation - but not always important

It depends on the dataset and the users' typical use cases how big of an issue this is.
Remember: if the underlying joint distribution does not change meaningfully, the curves will look mostly the same.

Dynamic and rebased SHAP DPs can yield almost identical results





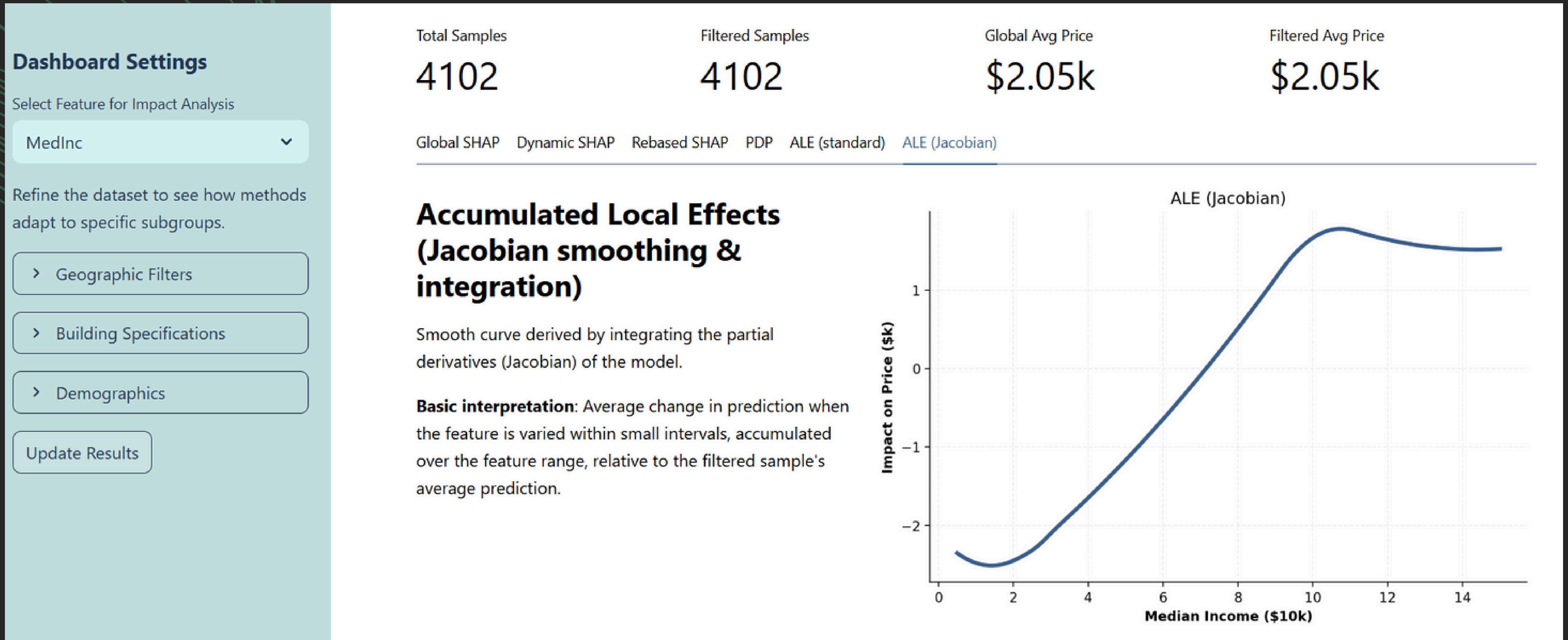
So what?

Interpretable ML for dashboards is often a trade-off between theorical correctness and UX.

What works in our notebooks can still break when users are left alone with the data.

A principled and statistically correct option may exist - but often there is a simple workaround.

And if you can’t get enough of this topic…



da-jamboree-mn-impact-curves.streamlit.app

…there is a dashboard for you!

Thank you - any questions?

Feel free to reach out on LinkedIn!

linkedin.com/in/nagymarton/