Interpretable Machine Learning, Part 4: SHAP

Shapely Values

Shapely values are a concept developed in 1951 in the field of game theory.

Big Idea: If you have a coalition of players all working together, and they achieve a certain payout, how do we divide that payout up into the individual contributions of each player?

That is, how important is each individual player?

Shapely Values

What is the connection to predictive models?

Let's say we are trying to predict the price of a house based on *sqft_living, grade, waterfront,* and *bedrooms*, and the average home price in our dataset is \$540,000.

We have a 3 bedroom, 3,560 square foot home, with grade 8, with waterfront 2, for which to model predicts a price of \$768,000.

Here the "payout" is the difference between our prediction and the average price (\$768,000 - \$540,000 = \$228,000)

We need to determine how to divide the \$228,000 difference among our 4 features.

Shapely Values - Mathematical Definition

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\;(n - |S| - 1)!}{n!}(v(S \cup \{i\}) - v(S))$$

Shapely value for feature *i*

Sum over all subsets of features that don't include feature *i*

Weight for that subset of features (large and small set of features get more weight)

Predicted value with feature *i*

Predicted value without feature *i*

Shapely Values

The formula from the previous slide involved calculating the prediction with and without certain features.

But, we cannot actually remove the feature, since the model expects us to pass in a value for it.

To “remove” the feature, we take the same approach as permutation importance: substitute in a random value instead.

If we repeat this a large number of times and take the average, we will have a pretty good approximation of the true Shapely value.

Shapely Values

Feature	Including Bedrooms	Bedrooms "Removed"
sqft_living	3560	3560
grade	8	8
waterfront	0	0
bedrooms	3	2
PREDICTION:	**$768,000**	**$787,000**
	MARGINAL CONTRIBUTION:	**-$19,000**

First, we substitute in random values for the number of bedrooms in order to "remove" that feature.

Shapely Values

Feature	Including Bedrooms	Bedrooms “Removed”
sqft_living	3560	3560
grade	8	8
waterfront	0	0
bedrooms	3	4
PREDICTION:	**$768,000**	**$1,045,00**
	MARGINAL CONTRIBUTION:	**-$277,000**

Shapely Values

Feature	Including Bedrooms	Bedrooms “Removed”
sqft_living	2480	2480
grade	8	8
waterfront	0	0
bedrooms	3	2
PREDICTION:	**$547,000**	**$727,000**
	MARGINAL CONTRIBUTION:	**-$180,000**

Here, we have “removed” the *sqft_living* feature

Shapely Values

Feature	Including Bedrooms	Bedrooms “Removed”
sqft_living	2480	2480
grade	8	8
waterfront	0	0
bedrooms	3	4
PREDICTION:	**$547,000**	**$557,000**
	MARGINAL CONTRIBUTION:	**-$10,000**

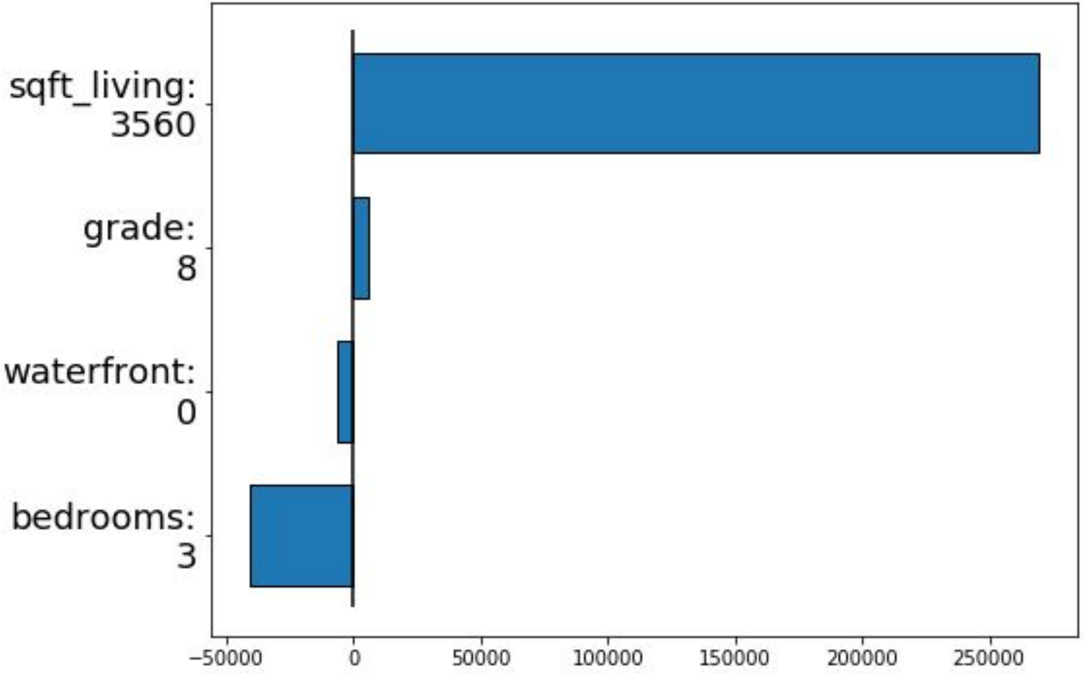
Shapely Values

Feature	Including Bedrooms	Bedrooms "Removed"
sqft_living	1500	1500
grade	9	9
waterfront	0	0
bedrooms	3	4
PREDICTION:	**$671,000**	**$772,000**
	MARGINAL CONTRIBUTION:	**-$101,000**

Here, we have "removed" the *sqft_living* and the *grade* features

Shapely Values

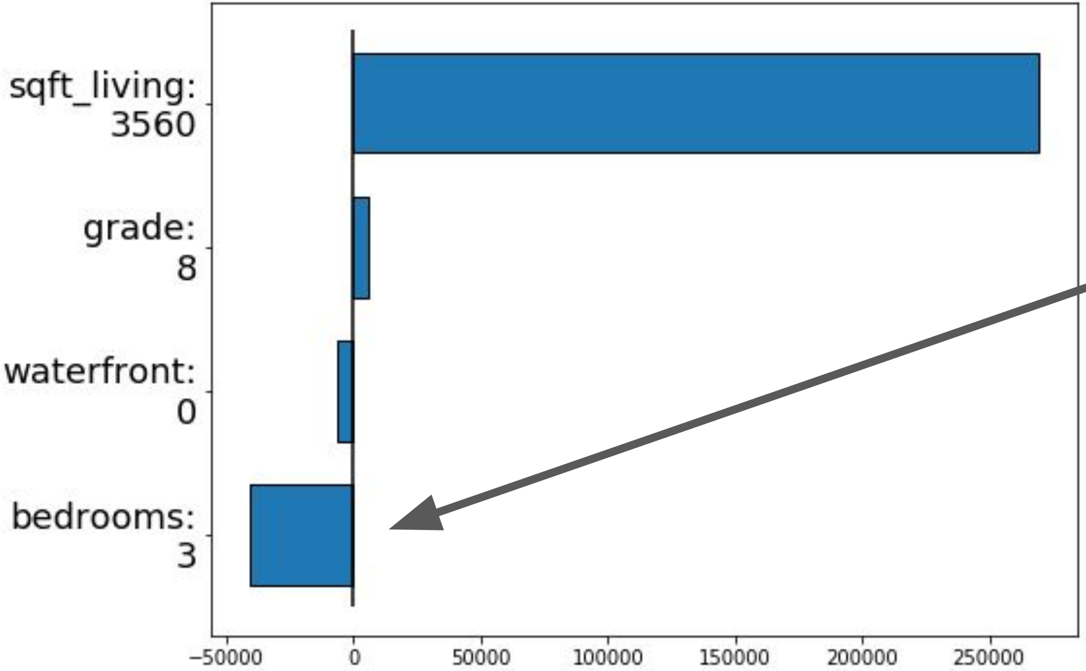
```
Average home price: 540088.14
Predicted home price: 768655.22
Difference from Average: 228567.08
```



Repeating this a large number of times and for each feature, we get this result.

Shapely Values

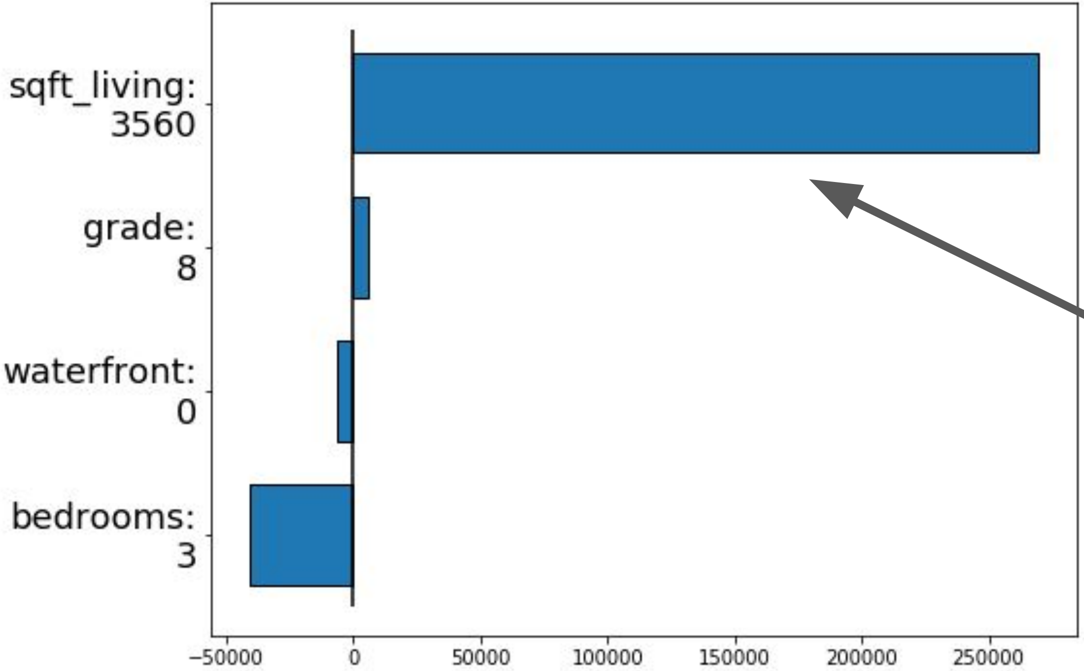
```
Average home price: 540088.14
Predicted home price: 768655.22
Difference from Average: 228567.08
```



The *bedrooms* feature decreased the prediction value by about $40,000

Shapely Values

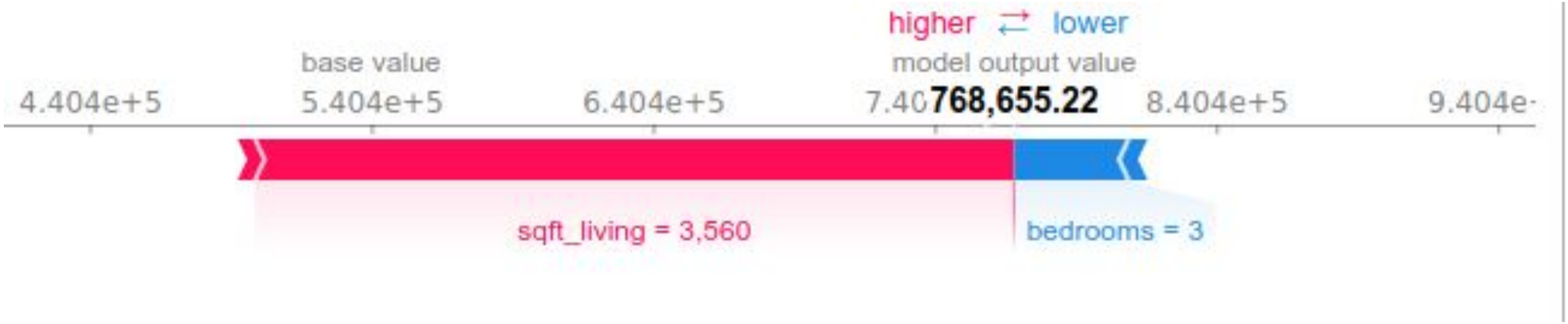
```
Average home price: 540088.14
Predicted home price: 768655.22
Difference from Average: 228567.08
```



The *sqft_living* feature increased the prediction value by about $270,000

Shapely Values

We can also see the results as a *force plot*, which takes the previous plot and displays it in a stacked format.



SHAP

In Python, we'll be using the *shap* library, which uses a method called SHAP (**SH**apely **A**dditive ex**P**lanations).

For each prediction, SHAP builds an "explanations model" *g*, where ϕ_j is the feature attribution for feature *j*, and z_j' indicates whether that feature is present or absent.

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z_j'$$

SHAP

For more details, see the original paper: https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

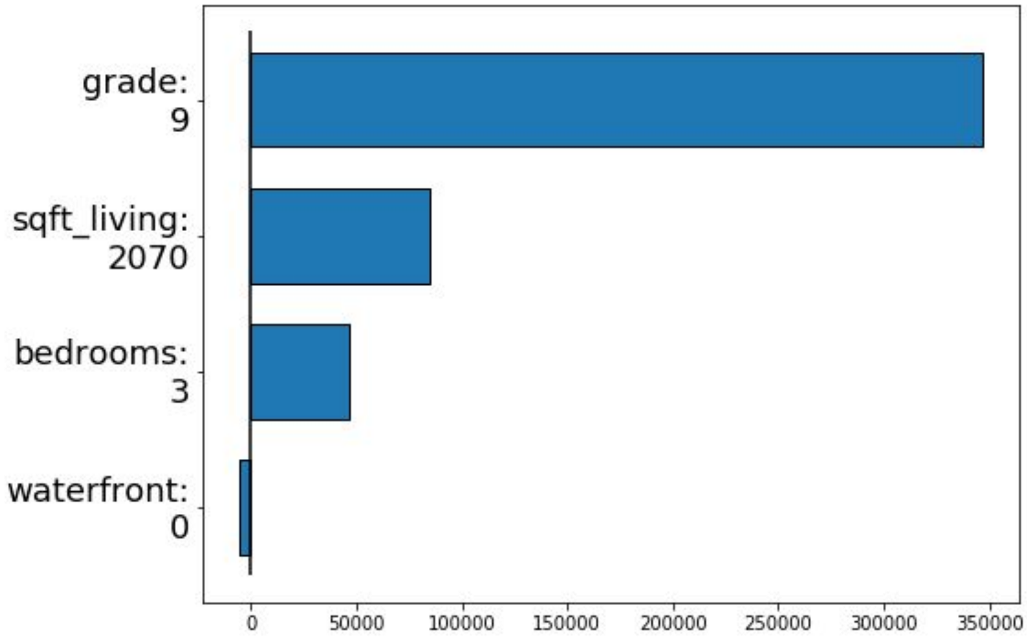
SHAP

There are two "flavors" of SHAP available in the *shap* library:

- KernelSHAP
  - This is essentially the procedure described above where we randomly replace certain features and see the effect on the predicted values
  - Computes an *estimate* of the Shapely values for each feature
- TreeSHAP
  - A variant of SHAP for tree-based models (random forests, and gradient-boosted trees, like XGBoost)
  - Computes *exact* Shapely values for each feature

SHAP Values

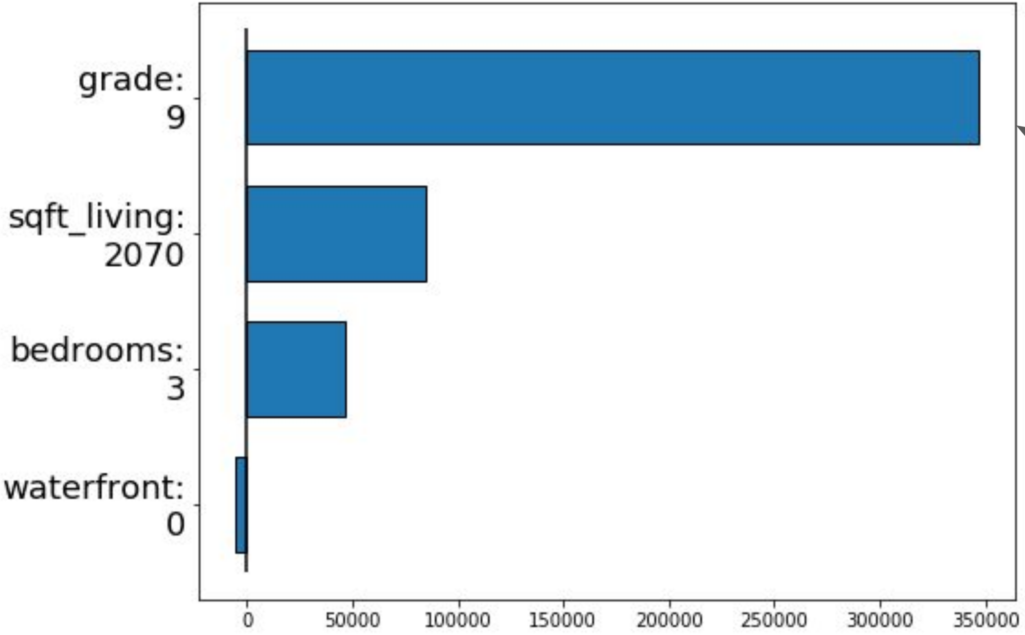
```
Average home price: 540088.14
Predicted home price: 1013793.76
Difference from Average: 473705.62
```



Another example. This time, a home with predicted price of $1,014,000. Where did the extra $470,000 come from?

SHAP Values

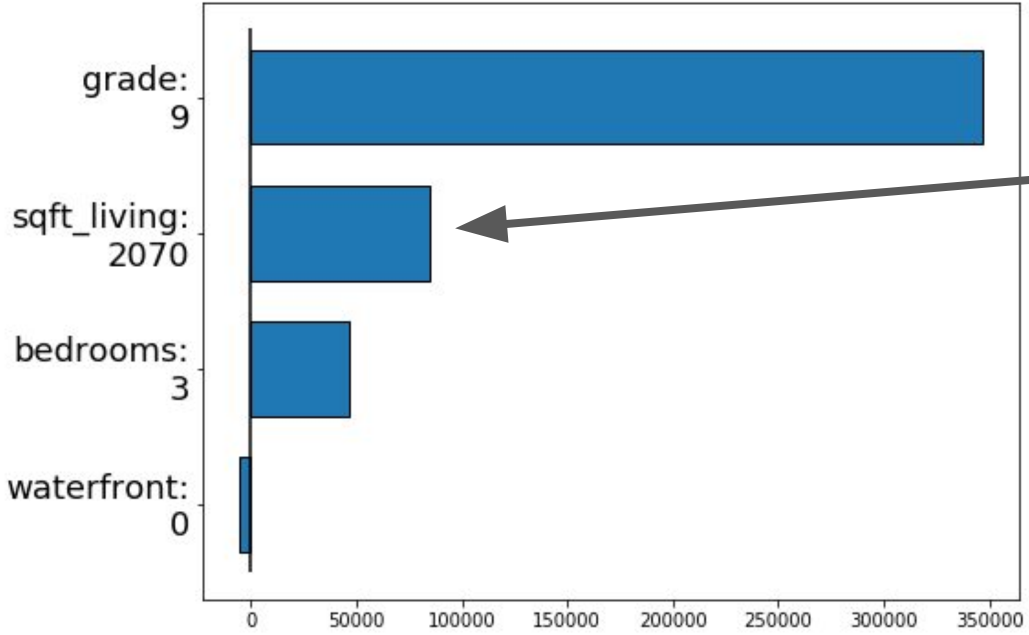
```
Average home price: 540088.14
Predicted home price: 1013793.76
Difference from Average: 473705.62
```



A great deal of it, almost $350,000, came from the fact that the grade was so high.

SHAP Values

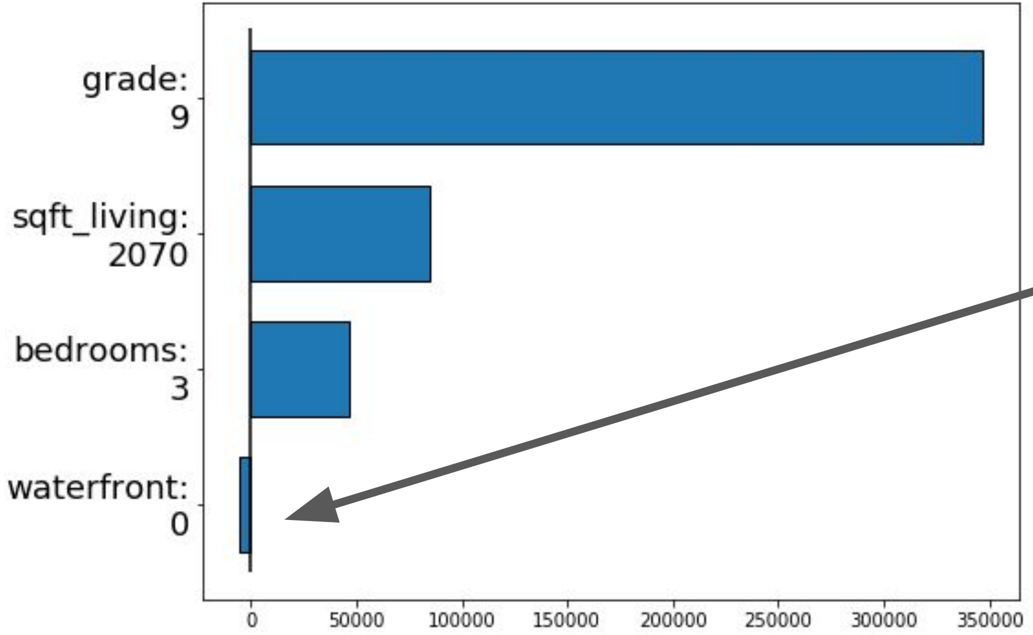
```
Average home price: 540088.14
Predicted home price: 1013793.76
Difference from Average: 473705.62
```



The living space added about $100,000

SHAP Values

```
Average home price: 540088.14
Predicted home price: 1013793.76
Difference from Average: 473705.62
```



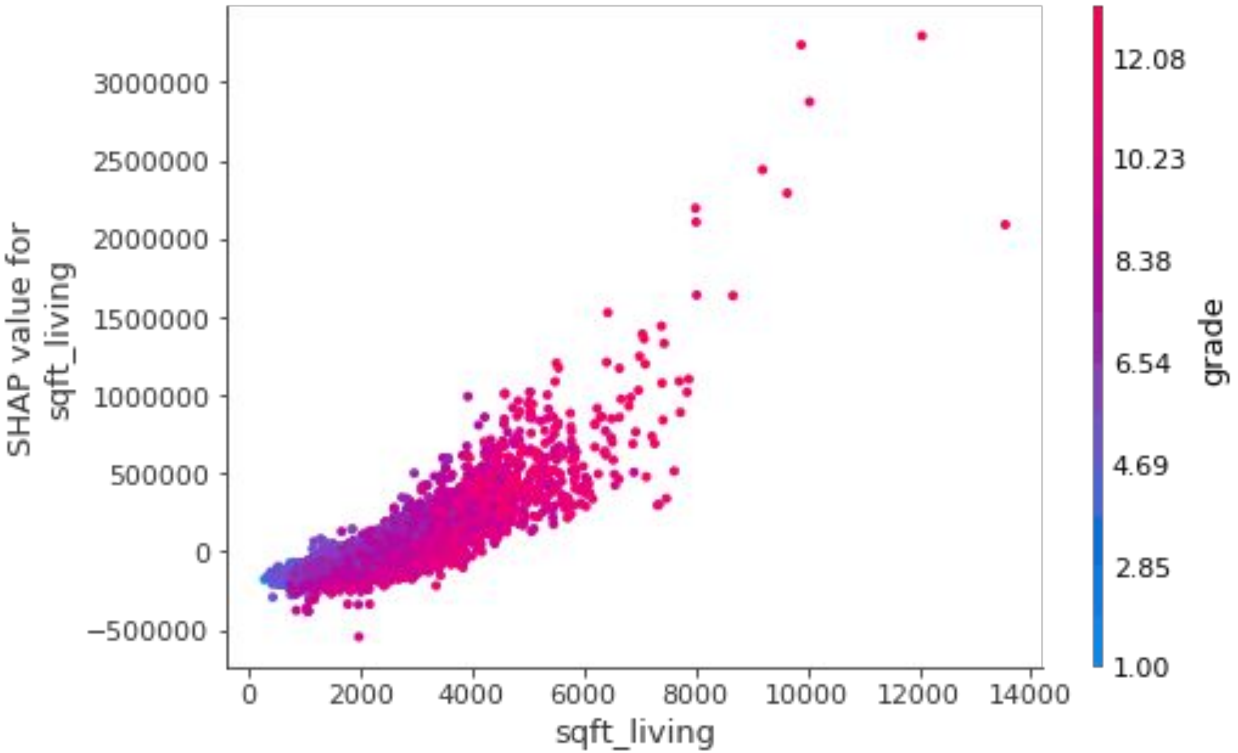
Since it was not a waterfront property, the predicted price dropped by about $5,000.

SHAP Values

We can also see the results as a *force plot*, which takes the previous plot and displays it in a stacked format.

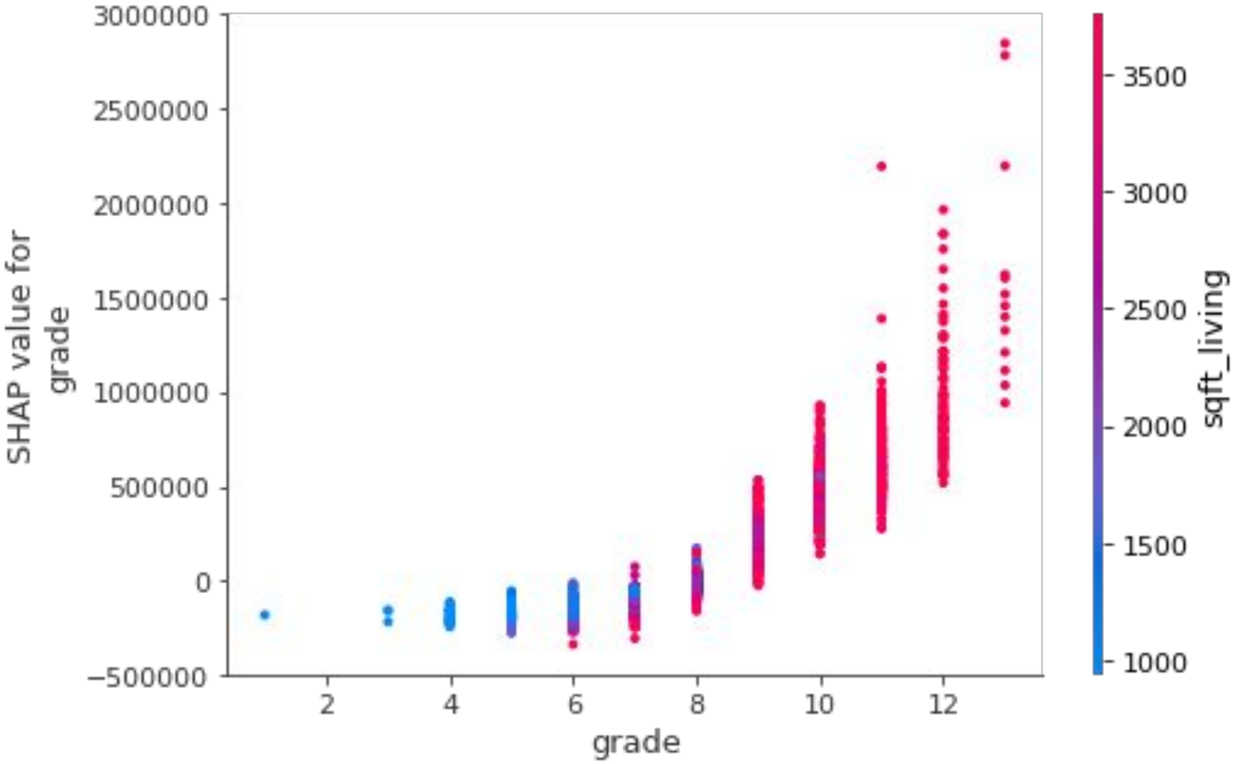
SHAP Values

We can also see how different features impact predictions across the entire dataset, similarly to a partial dependence plot.



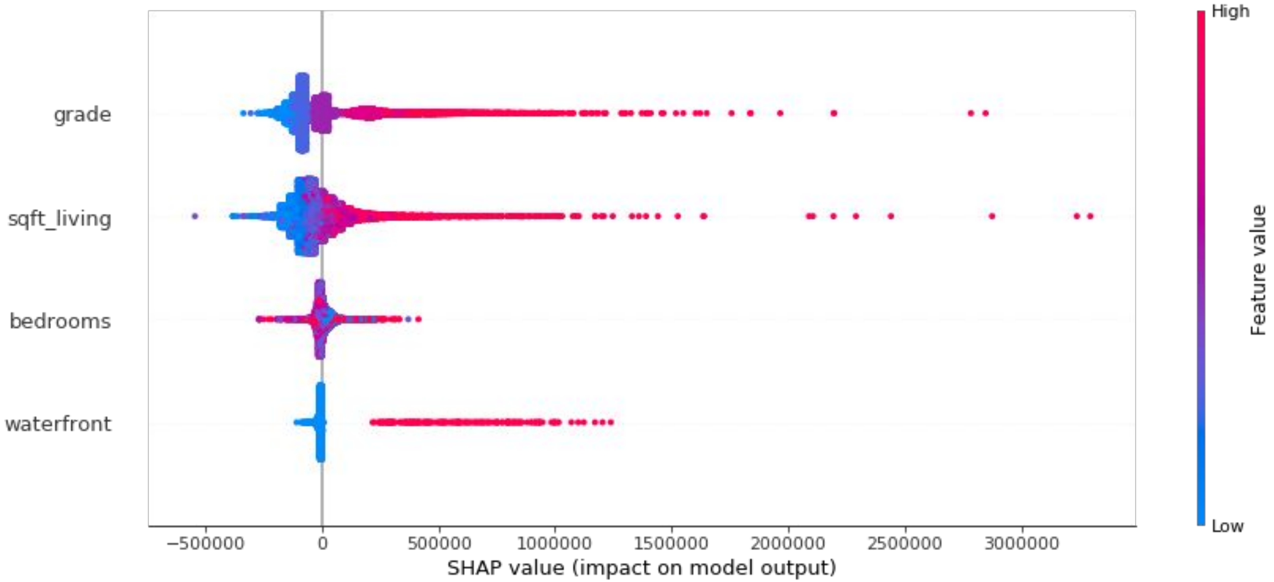
SHAP Values

We can also see how different features impact predictions across the entire dataset, similarly to a partial dependence plot.



SHAP Values

Or see the effect of each feature across the entire dataset.



SHAP Values

Or see the effect of each feature across the entire dataset.