# Interpretable Machine Learning Part 2: Importance Scores

# Understanding Importance Scores

One way to understand how a model works is to understand which features the model relies on to make predictions.

An **importance score** quantifies how much reliance a model has on a particular feature.

Importance scores do not usually indicate *how* a model uses a feature, only that it is important. That is, we won't know if increasing that feature will increase or decrease the response variable.

# Decision Tree Feature Importance

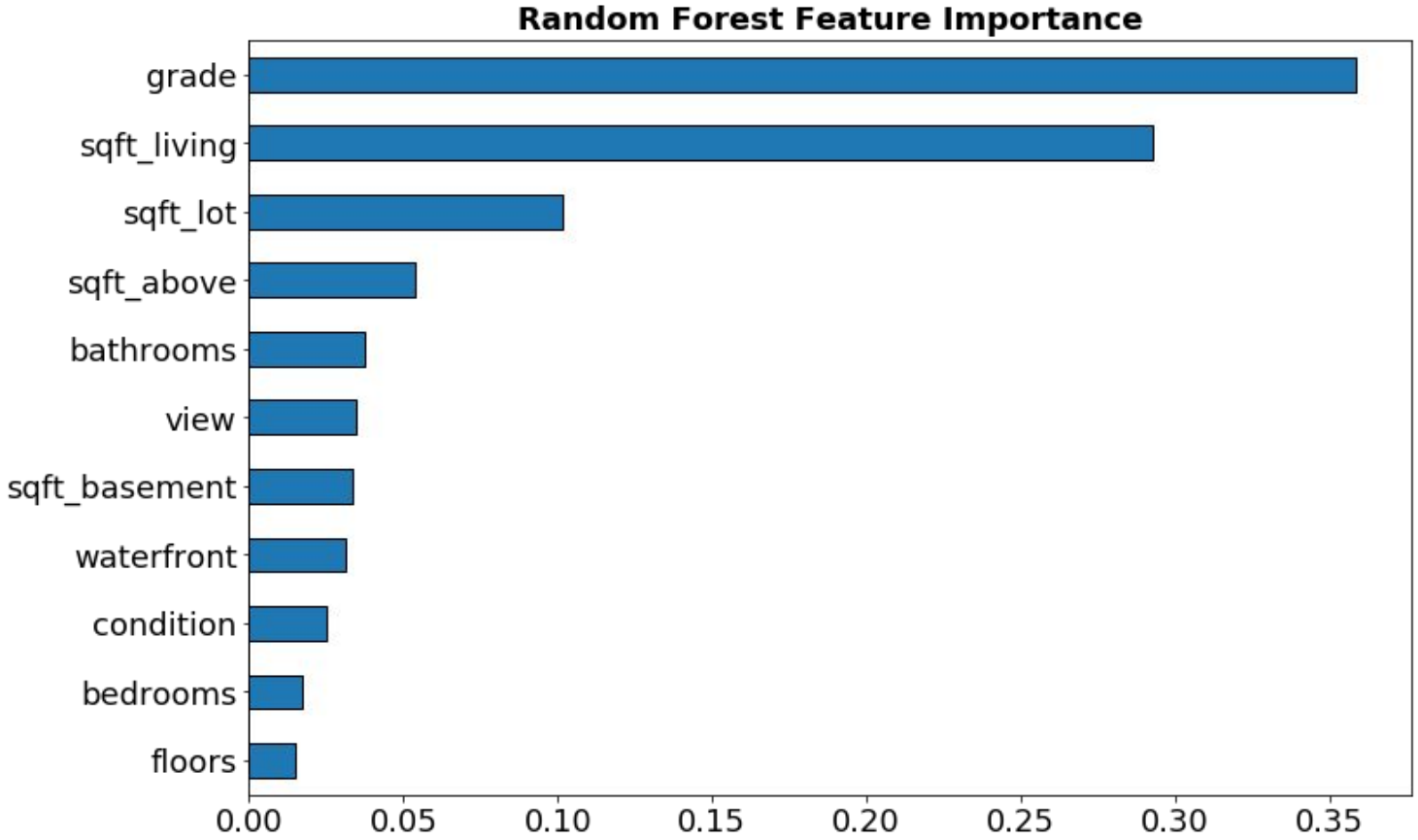
Decision Tree models and decision tree based models (random forests, gradient-boosted trees) have built-in importance metrics.

This is usually calculated as the weighted (by the number of observations in the node) information gain from using that feature to split the tree.

The amount of “information” in a node can be calculated as the Gini Impurity (https://bambielli.com/til/2017-10-29-gini-impurity/) for classification problems or variance for regression.

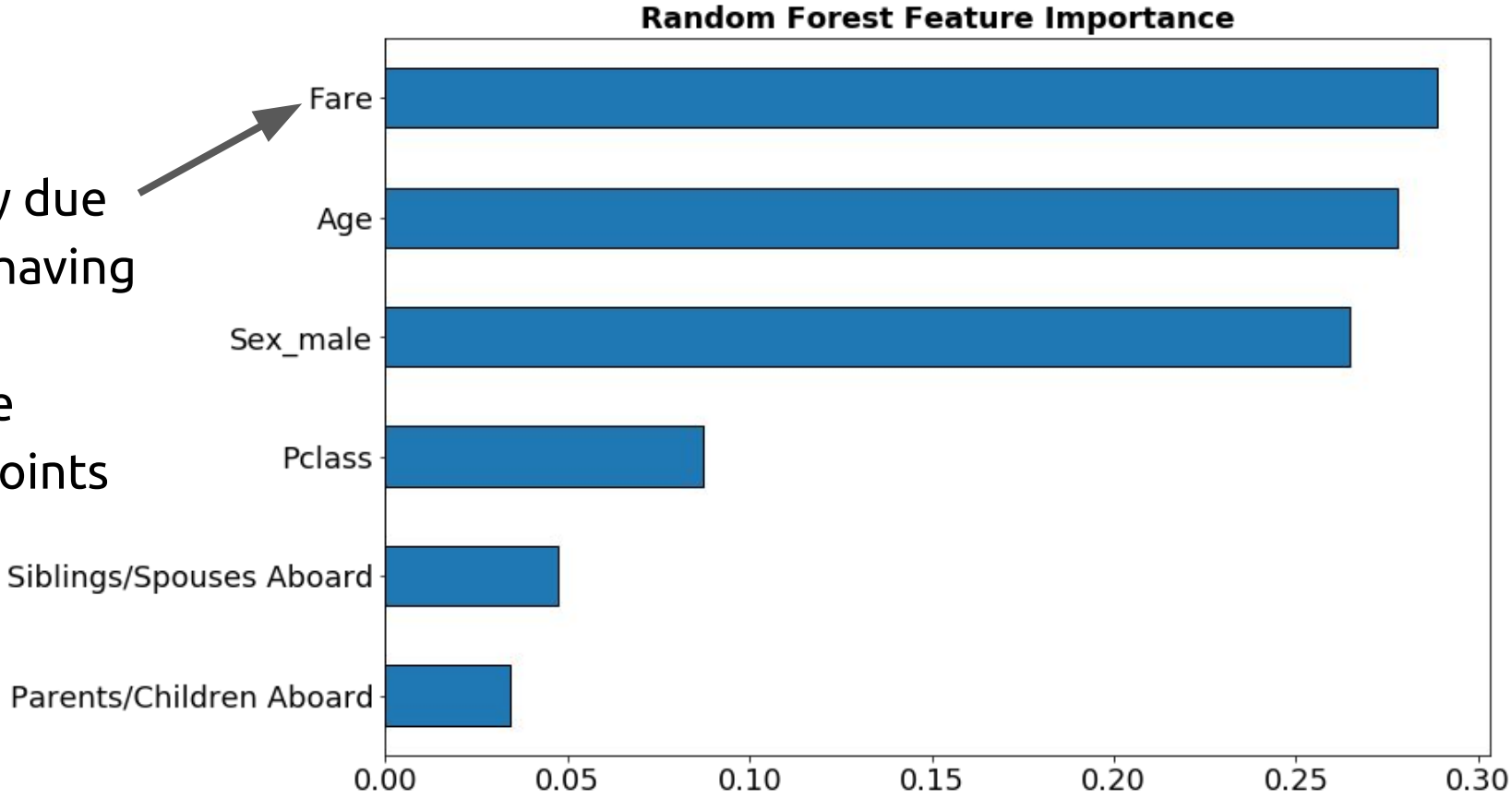
**Warning:** This can be biased to features that have a wider range or number of categories, since they have more opportunity to be split!

# Decision Tree Feature Importance: Housing



# Decision Tree Feature Importance: Titanic

Possibly due to *fare* having a lot of possible break points



# Permutation Importance

While decision tree based models have a built-in feature importance, we can also calculate feature importance in a model-agnostic way.

For a trained model, we want to see how much each feature contributes to the performance of the model.

**Idea:** Take your *trained* model and “cancel out” a feature at a time.

How? We can’t simply delete that feature, because the model is already trained and expects you to input that feature.

Instead, we make the feature of interest “random noise” by permuting the original values.

# Permutation Importance

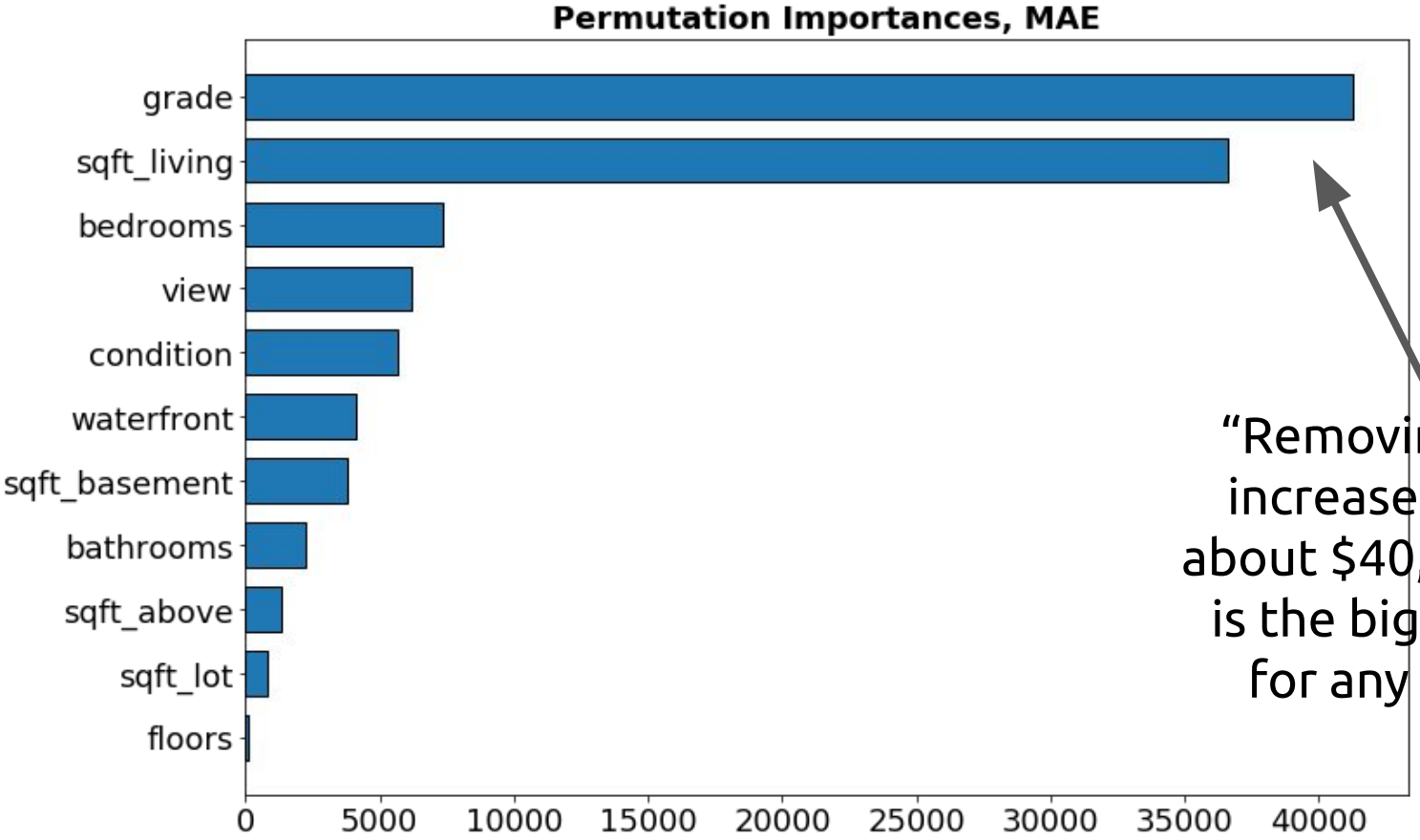
Then we compare the prediction accuracy on the actual data to the data with the feature of interest randomly permuted.

If there is a large drop, then that feature is deemed to be “important”.

If the drop in performance is small, then that feature must have not been very important to the model’s performance.

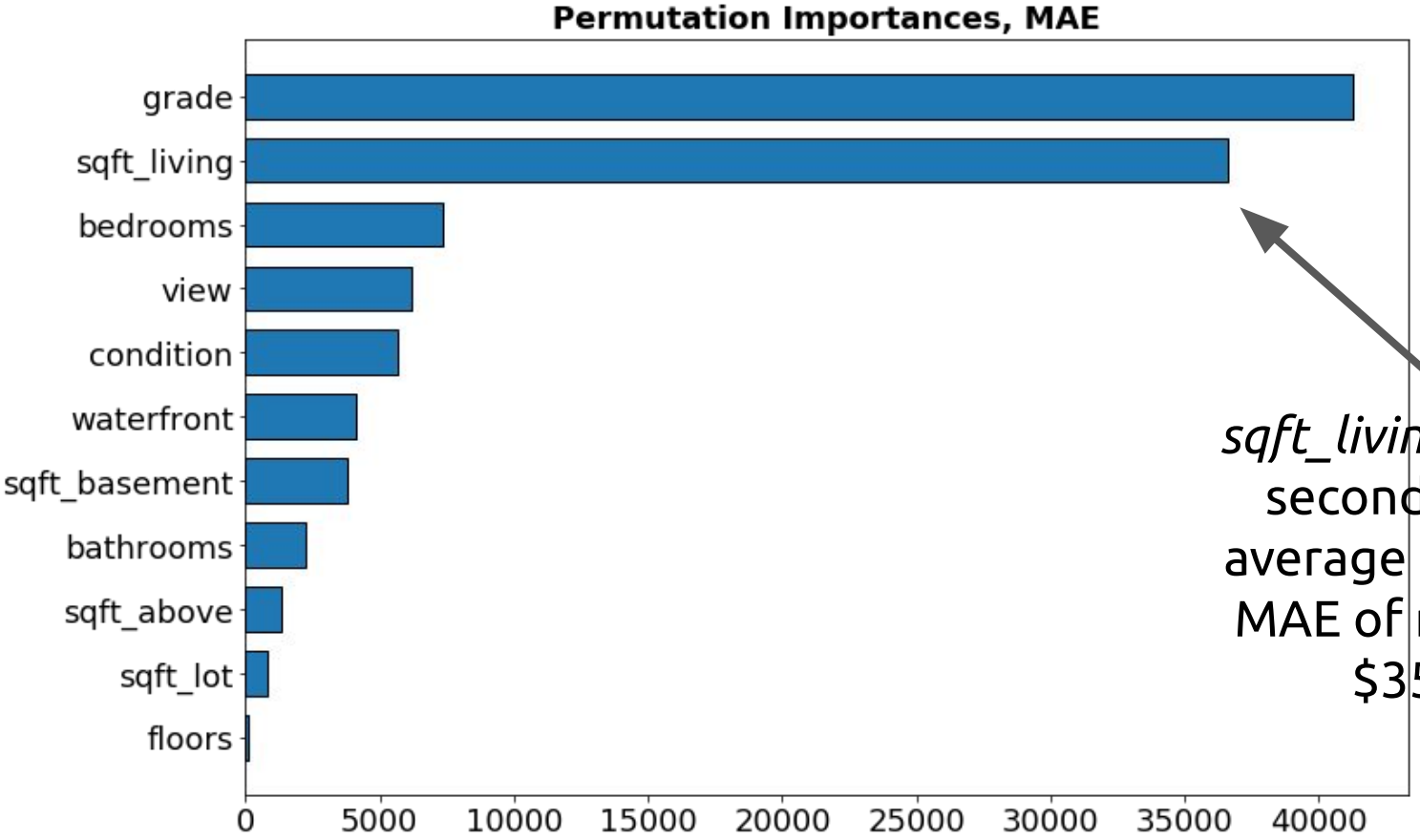
This can be done using the *permutation_importance* function from scikit-learn’s inspection module.

# Permutation Importance - Housing Model



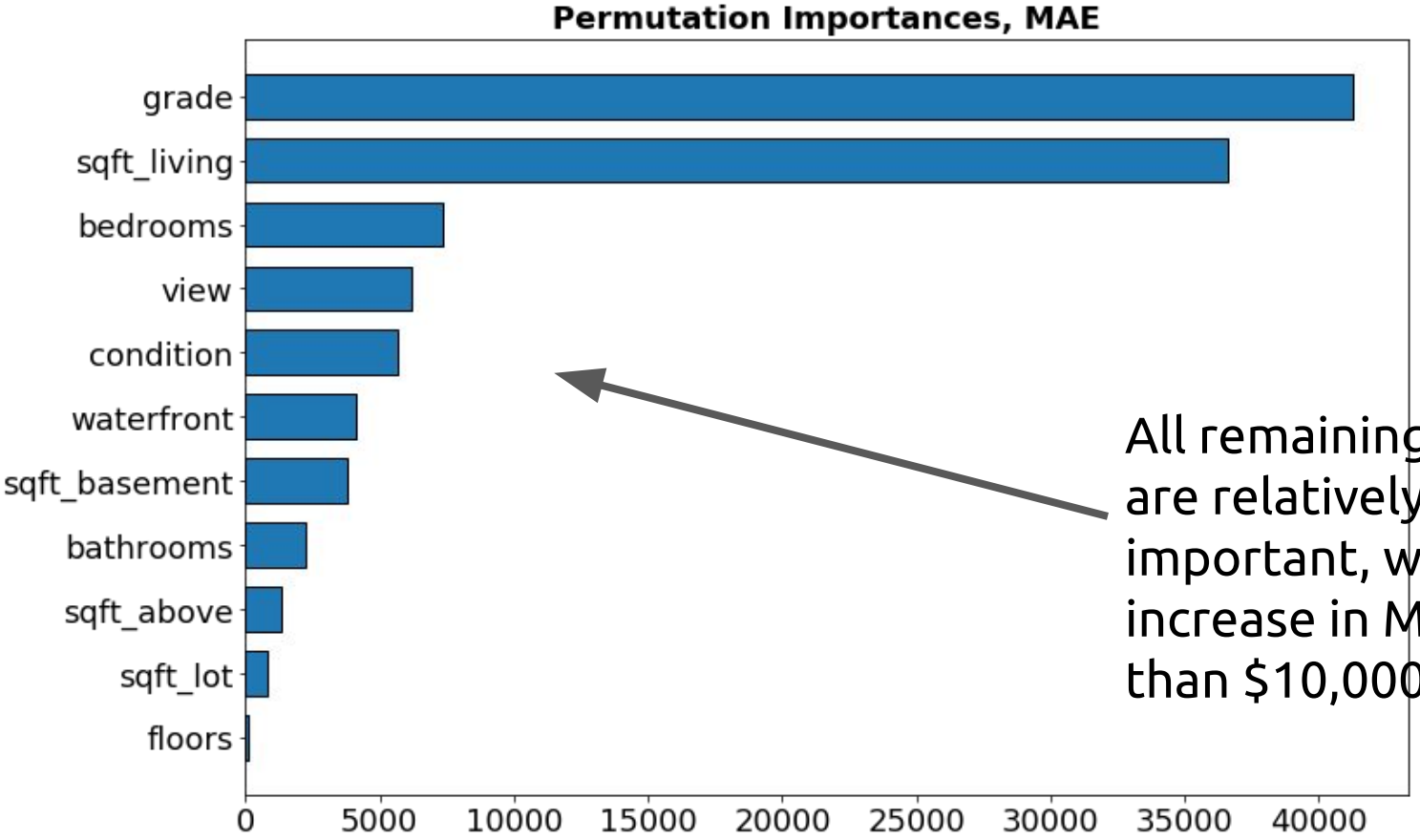

"Removing" *grade* increases MAE by about $40,000, which is the biggest drop for any feature.

# Permutation Importance - Housing Model



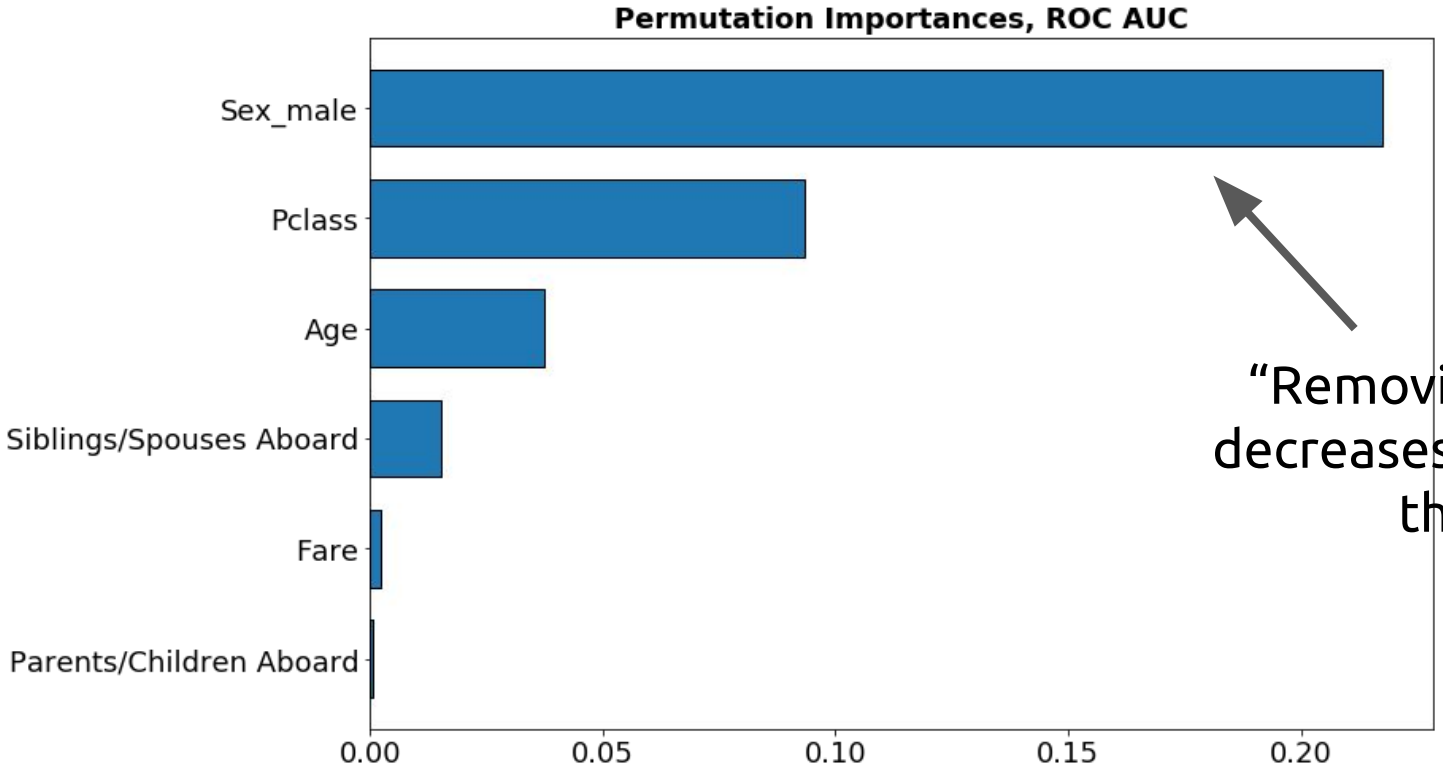

*sqft_living* is a close second, with an average increase in MAE of more than $35,000

# Permutation Importance - Housing Model



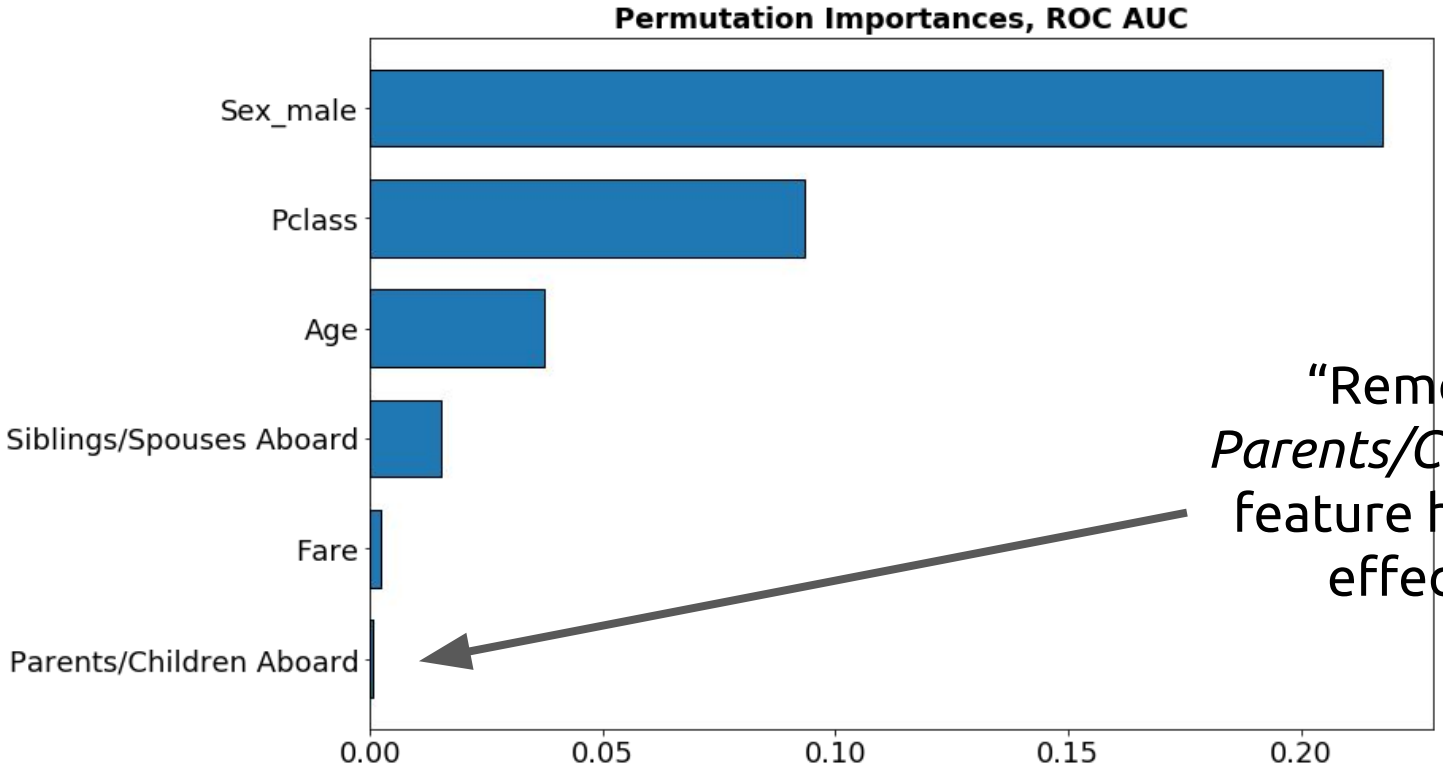

All remaining features are relatively less important, with an increase in MAE of less than $10,000.

# Permutation Importance - Titanic Model




"Removing *Sex_male* decreases AUC by more than 0.2

# Permutation Importance - Titanic Model




"Removing" the *Parents/Children Aboard* feature has almost no effect on AUC

# Permutation Importance

**Advantages:**

- Can be used on any type of model
- Fast and easy to implement
- Returns an importance score in terms of any metric you are interested in

**Disadvantages:**

- Can force model to “extrapolate” to regions of the input space with no real observations (what if two of our features are male/female and pregnant/not pregnant)
- If you have correlated but important features, they can end up with low permutation importance scores, especially for flexible models (since the effect  of these features can be split between them)

See this article for a more in-depth discussion of the downsides to permutation importance: https://arxiv.org/pdf/1905.03151.pdf

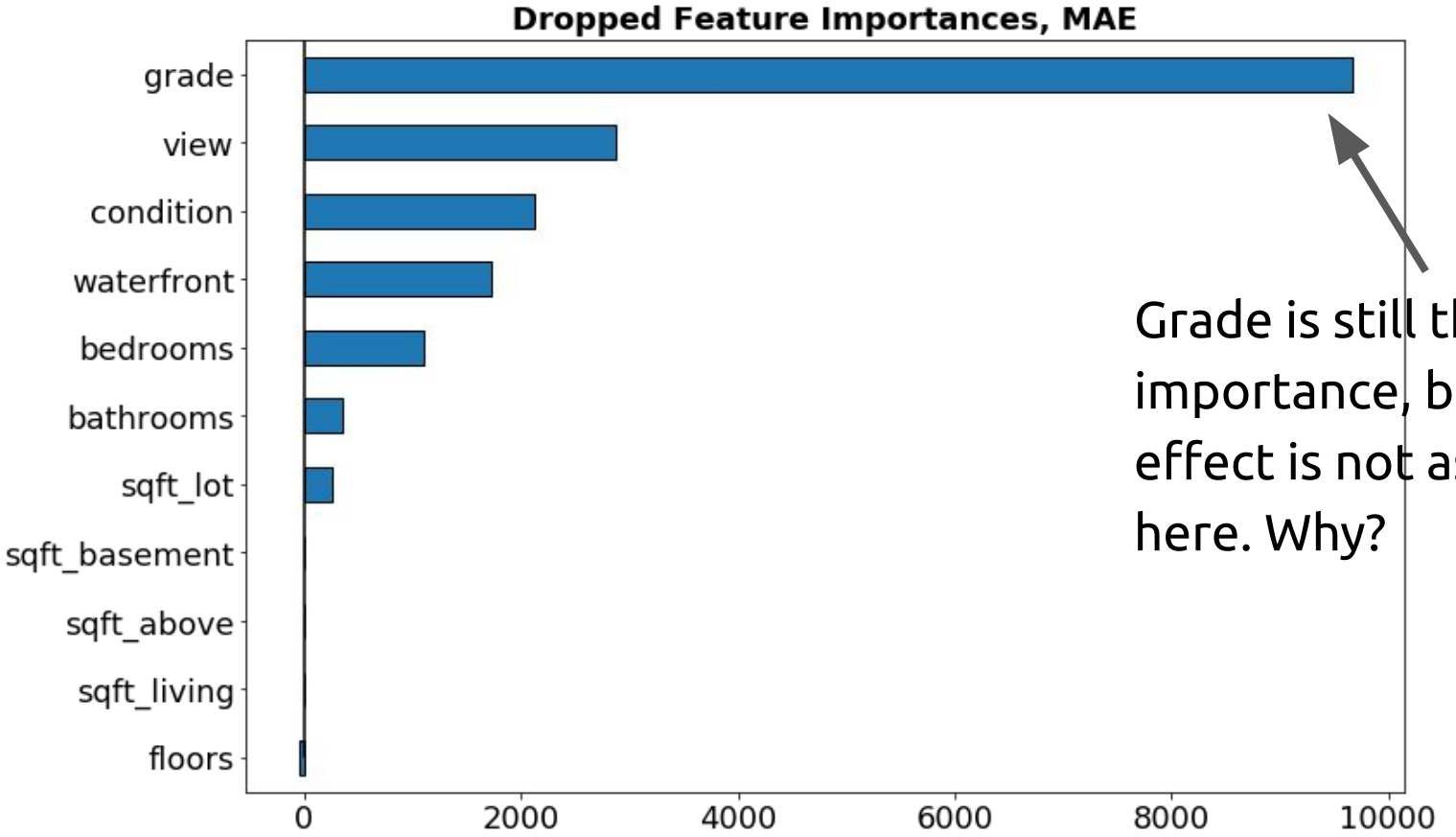
# Dropped Feature Importance

Rather than just permuting a variable, drop that variable and retrain the model to see the drop in performance.

Much slower than permutation importance since it requires retraining the model for each feature.

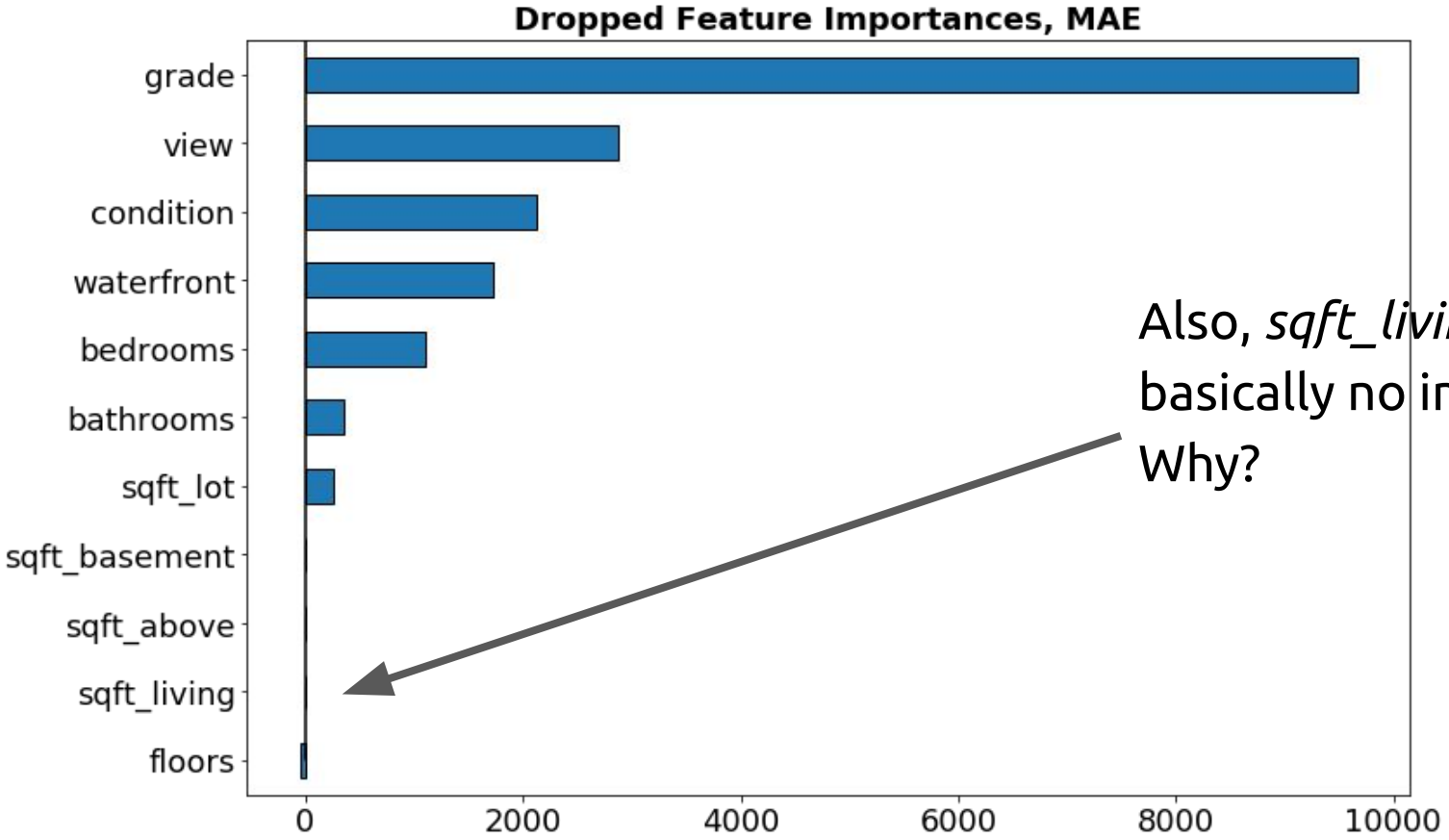
Also, correlated features can throw off dropped feature importance, since the effect of a feature can be captured by the features it is correlated with, without having model performance suffer by much.

# Dropped Feature Importance - Housing Model



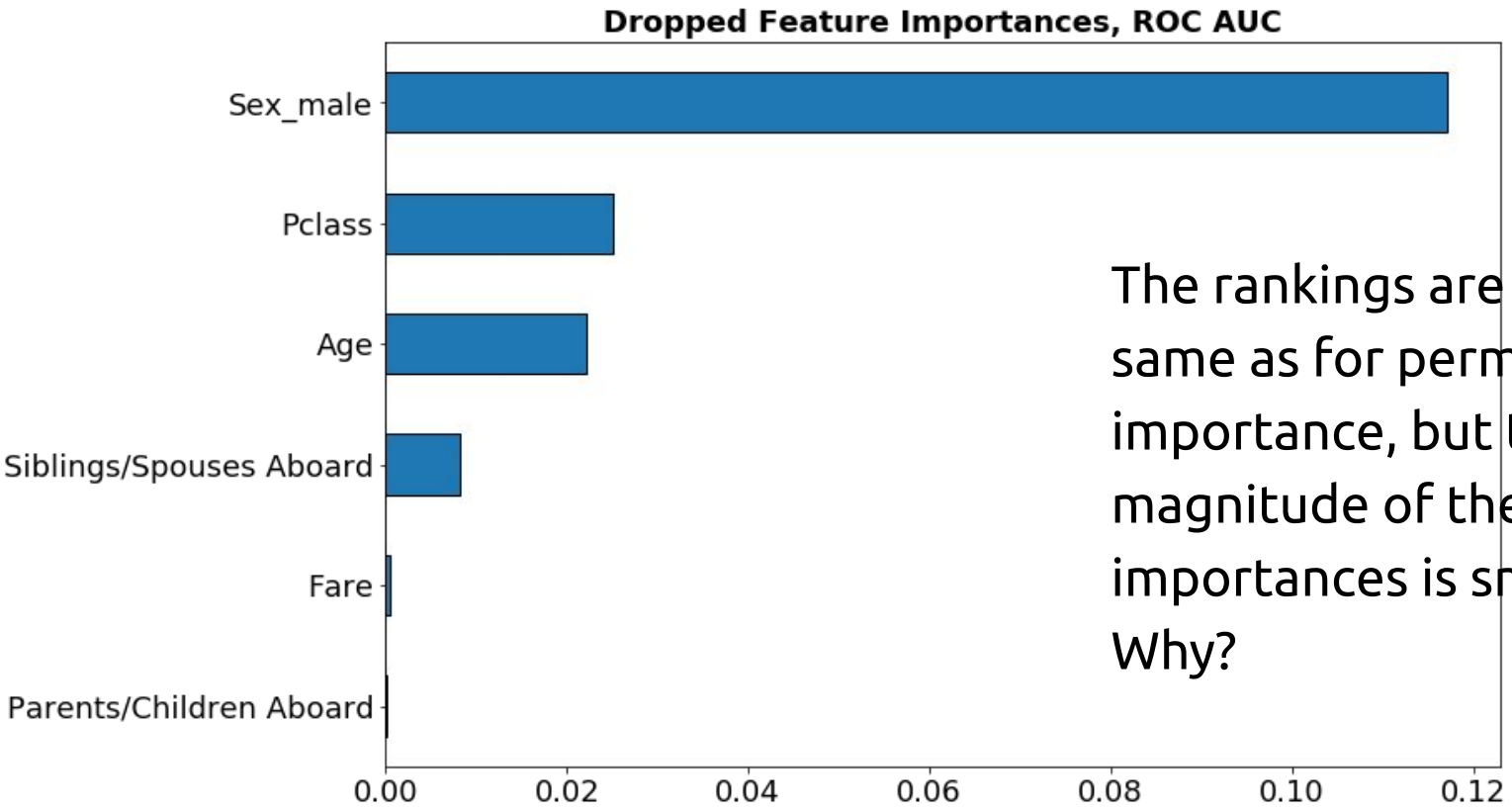

Grade is still the highest importance, but it's effect is not as strong here. Why?

# Dropped Feature Importance - Housing Model




Also, *sqft_living* has basically no importance. Why?

# Dropped Feature Importance - Titanic Model




The rankings are the same as for permutation importance, but the magnitude of the importances is smaller. Why?