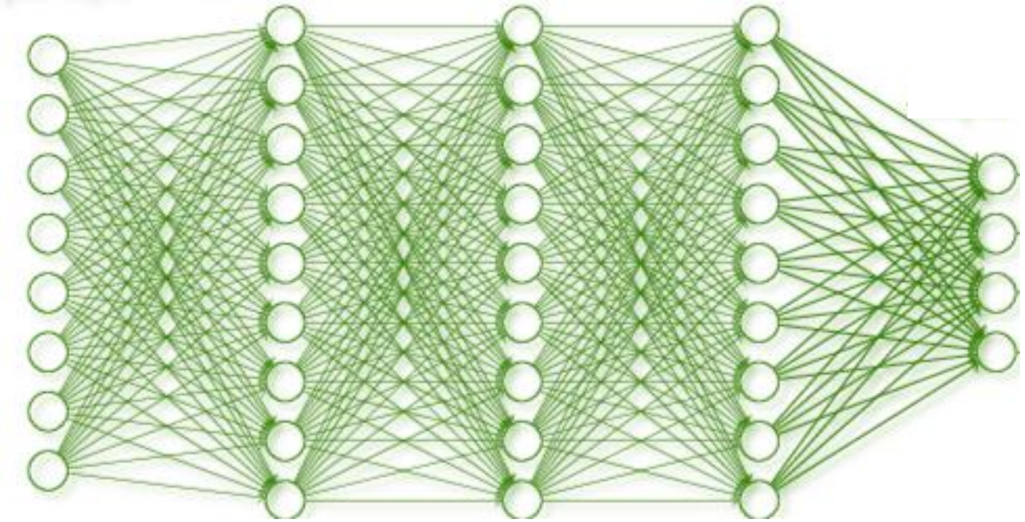
# The LIME Package

James Lloyd

# The Problem Statement

- Deep learning tools and algorithms are powerful, but not completely trusted – we like to know what's going on.
- Netflix does not make use of neural networks or deep learning
- Programmers at the University of Washington decided to tackle the 'Black Box' problem of modern machine learning techniques.



*You* explain this thing!

James Lloyd

# The LIME Package

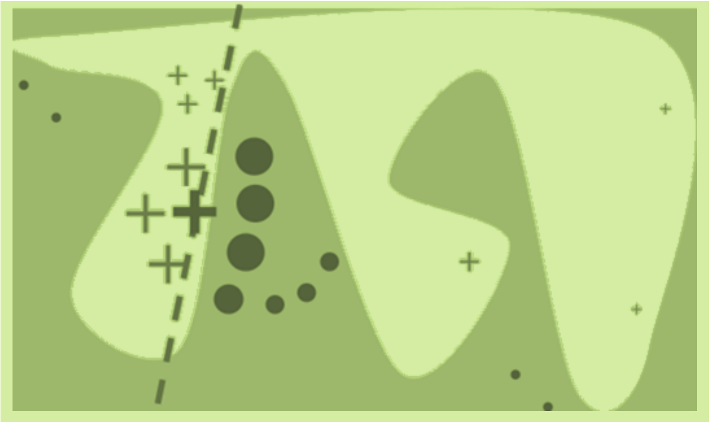
- Their solution was the **LIME** Package, an acronym for the desired characteristics of an explainer:
  - **LOCAL**
  - **INTERPRETABLE**
  - **MODEL-AGNOSTIC**
  - **EXPLANATIONS**

## Model Explanations


James Lloyd

# Local

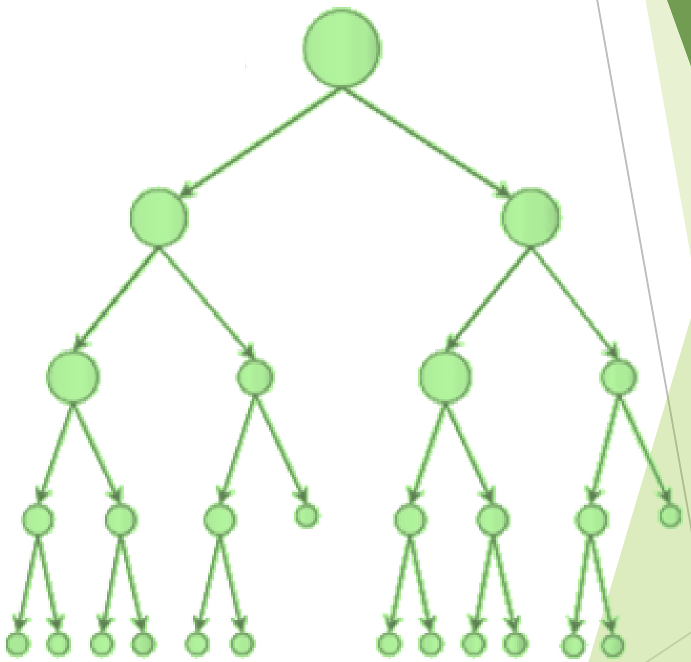
- Given the complexities of deep learning algorithms, a universal explanation of the model is either not be possible or not comprehensible.
- The solution was to explain model behaviour around a particular point, which would be fairly robust.
- Consider the example below:



- The region around the + is essentially linearly separable, and the variables responsible for this separation can be examined.

# Interpretable

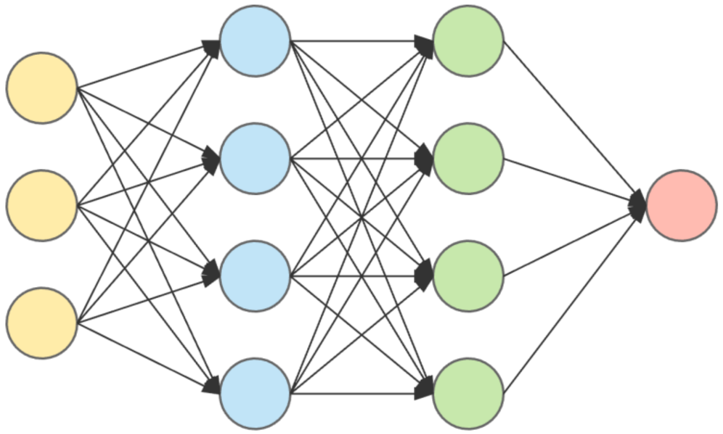
- Decision trees and regression models are popular because they are easy to explain and understand.
- The effects of each variable can be quantified
- The creators of **LIME** leveraged the interpretability of these simpler models to build their explainer
- An interpretable model is built on permutations around the point that needs explaining



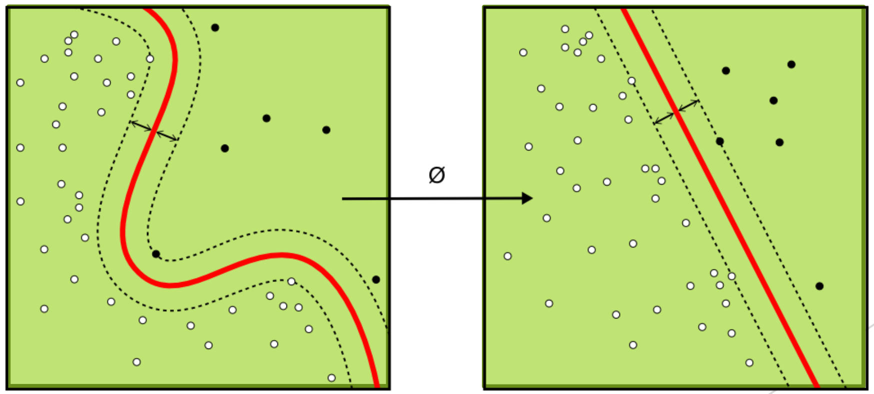
# Model-Agnostic

- The creators of **LIME** didn’t want users to restricted to particular models to use their package – flexibility was a main consideration.

- The assumption that linear approximations are generally valid in the proximity of the observation allows the package to explanations for predictions made using any model – linear models are simply built around that point.



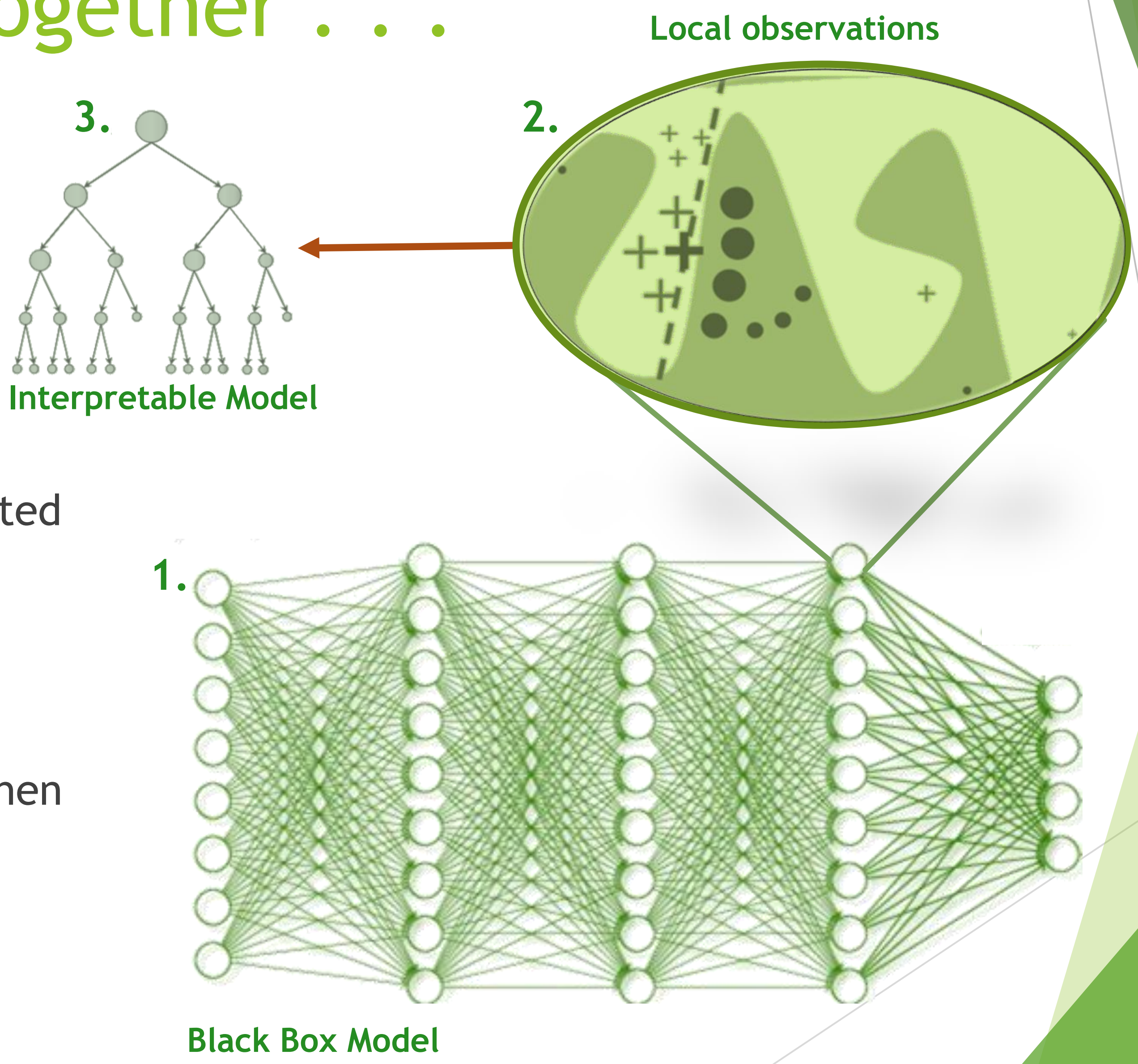
Neural Network




Support Vector Machine

# Putting it all together . . .

- A random permutation of points are generated around the observation
- The observations are weighted (usually exponentially) according to their distance from the point of interest.
- An interpretable model is then built to predict the transformed observations.

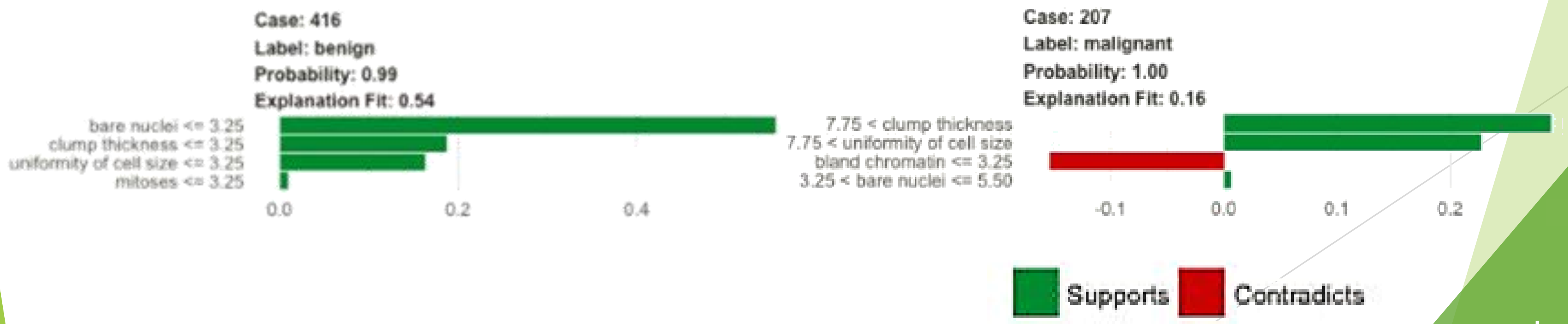



James Lloyd

# What the code looks like

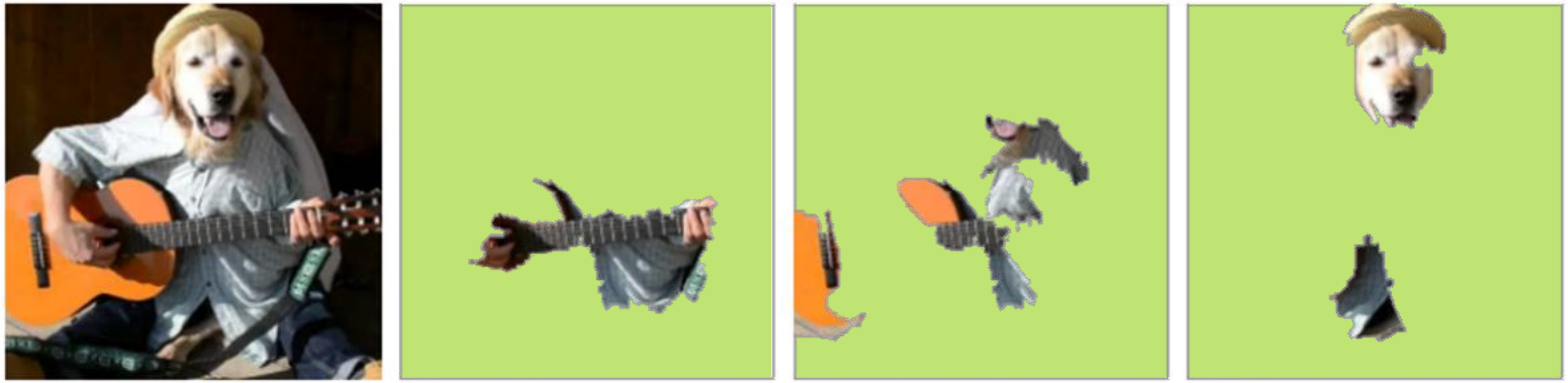
- Using the **LIME** package requires you define an explainer, and use it generate the explanation.
- The explainer takes in the training data and the model.
- The explanation uses the explainer to explain the observations witnessed on an entirely new data set, identifying the *n_features* most important variables.

```
explainer <- lime(biopsy[-test_set,], model, bin_continuous = TRUE, quantile_bins = FALSE)
explanation <- explain(biopsy[test_set, ], explainer, n_labels = 1, n_features = 4)
plot_features(explanation, ncol = 1)
```




James Lloyd

# Examples from Image Classification


(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

- A neural net was used to classify the contents of the image above
- The results were *Electric guitar*, *Acoustic guitar* and *Labrador*
- Running this through the **LIME** package resulted in thee important features for each of the classifications
  - Which is pretty damn cool!

James Lloyd

# Examples . . . gone wrong

- **LIME** is certainly still a work in progress, as shown in the image below (a gentle reminder that black box algorithms are still rather . . . black box)
- Regardless, **LIME** provides a powerful means for better understanding your model and convincing business of the value in your work.

(a) Husky classified as wolf (b) Explanation