# Interpretable Machine Learning (including Deep Learning)

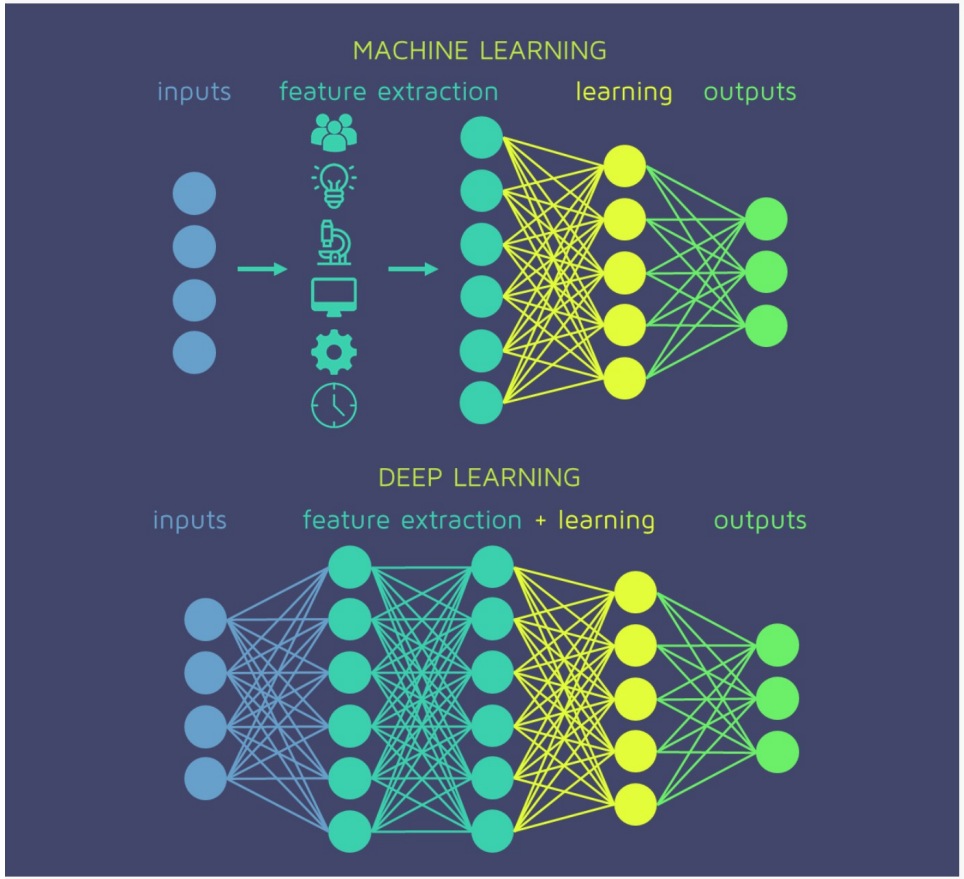
## With Applications

Natalia Connolly
November 2019

# Outline

- Machine learning vs. deep learning - what’s the difference?
- Why data science model interpretability is important
- Three main approaches to interpretability:
  - Model-agnostic
  - Model-specific
  - Out of the box interpretable models
- Conclusions

# Machine Learning vs. Deep Learning



# Machine Learning vs. Deep Learning (cont.)

- Machine Learning:
  - Relies on “human in the middle” for
    - Careful feature engineering
    - Domain expertise
  - Typically does not require “extreme” computational resources (e.g., “laptop-friendly”)

- Deep Learning:
  - Automatically learn features needed for classification of detection using raw data
  - Often requires large datasets and significant computational resources

# Our use case: churn

- We will use a telecom company's called *Telco* customer churn dataset
  - 7,032 customers
  - 20 features
    - Tenure, demographics, types of services, multiple lines of service, payment method, charges…
- The goal is to predict whether a customer will churn or not
  - Training data has “churn” indicators - whether or not a customer left the company within the last month

| gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | OnlineBackup | DeviceProtection | TechSupport | StreamingTV | StreamingMovies | Contract | PaperlessBilling | PaymentMethod | MonthlyCharges | TotalCharges | Churn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | Yes | No | No | No | No | Month-to-month | Yes | Electronic check | 29.85 | 29.85 | No |
| Male | 0 | No | No | 34 | Yes | No | DSL | Yes | No | Yes | No | No | No | One year | No | Mailed check | 56.95 | 1889.50 | No |
| Male | 0 | No | No | 2 | Yes | No | DSL | Yes | Yes | No | No | No | No | Month-to-month | Yes | Mailed check | 53.85 | 108.15 | Yes |
| Male | 0 | No | No | 45 | No | No phone service | DSL | Yes | No | Yes | Yes | No | No | One year | No | Bank transfer (automatic) | 42.30 | 1840.75 | No |
| Female | 0 | No | No | 2 | Yes | No | Fiber optic | No | No | No | No | No | No | Month-to-month | Yes | Electronic check | 70.70 | 151.65 | Yes |
| Female | 0 | No | No | 8 | Yes | Yes | Fiber optic | No | No | Yes | No | Yes | Yes | Month-to-month | Yes | Electronic check | 99.65 | 820.50 | Yes |
| Male | 0 | No | Yes | 22 | Yes | Yes | Fiber optic | No | Yes | No | No | Yes | No | Month-to-month | Yes | Credit card (automatic) | 89.10 | 1949.40 | No |
| Female | 0 | No | No | 10 | No | No phone service | DSL | Yes | No | No | No | No | No | Month-to-month | No | Mailed check | 29.75 | 301.90 | No |
| Female | 0 | Yes | No | 28 | Yes | Yes | Fiber optic | No | No | Yes | Yes | Yes | Yes | Month-to-month | Yes | Electronic check | 104.80 | 3046.05 | Yes |
| Male | 0 | No | Yes | 62 | Yes | No | DSL | Yes | Yes | No | No | No | No | One year | No | Bank transfer (automatic) | 56.15 | 3487.95 | No |
| Male | 0 | Yes | Yes | 13 | Yes | No | DSL | Yes | No | No | No | No | No | Month-to-month | Yes | Mailed check | 49.95 | 587.45 | No |
| Male | 0 | No | No | 16 | Yes | No | No | No internet service | No internet service | No internet service | No internet service | No internet service | No internet service | Two year | No | Credit card (automatic) | 18.95 | 326.80 | No |
| Male | 0 | Yes | No | 58 | Yes | Yes | Fiber optic | No | No | Yes | No | Yes | Yes | One year | No | Credit card (automatic) | 100.35 | 5681.10 | No |
| Male | 0 | No | No | 49 | Yes | Yes | Fiber optic | No | Yes | Yes | No | Yes | Yes | Month-to-month | Yes | Bank transfer (automatic) | 103.70 | 5036.30 | Yes |
| Male | 0 | No | No | 25 | Yes | No | Fiber optic | Yes | No | Yes | Yes | Yes | Yes | Month-to-month | Yes | Electronic check | 105.50 | 2686.05 | No |
| Female | 0 | Yes | Yes | 69 | Yes | Yes | Fiber optic | Yes | Yes | Yes | Yes | Yes | Yes | Two year | No | Credit card (automatic) | 113.25 | 7895.15 | No |
| Female | 0 | No | No | 52 | Yes | No | No | No internet service | No internet service | No internet service | No internet service | No internet service | No internet service | One year | No | Mailed check | 20.65 | 1022.95 | No |
| Male | 0 | No | Yes | 71 | Yes | Yes | Fiber optic | Yes | No | Yes | No | Yes | Yes | Two year | No | Bank transfer (automatic) | 106.70 | 7382.25 | No |

# Start with a simple, interpretable models

- Many simple ML models are “naturally” interpretable
  - E.g., linear/logistic regression
  - In logistic regression, we calculate the (log) odds:

$$\frac{P_{churned}}{(1 - P_{churned})}$$

  - ...and we assume it’s a linear function of various regressors (features)

# Churn: Logistic Regression, one regressor

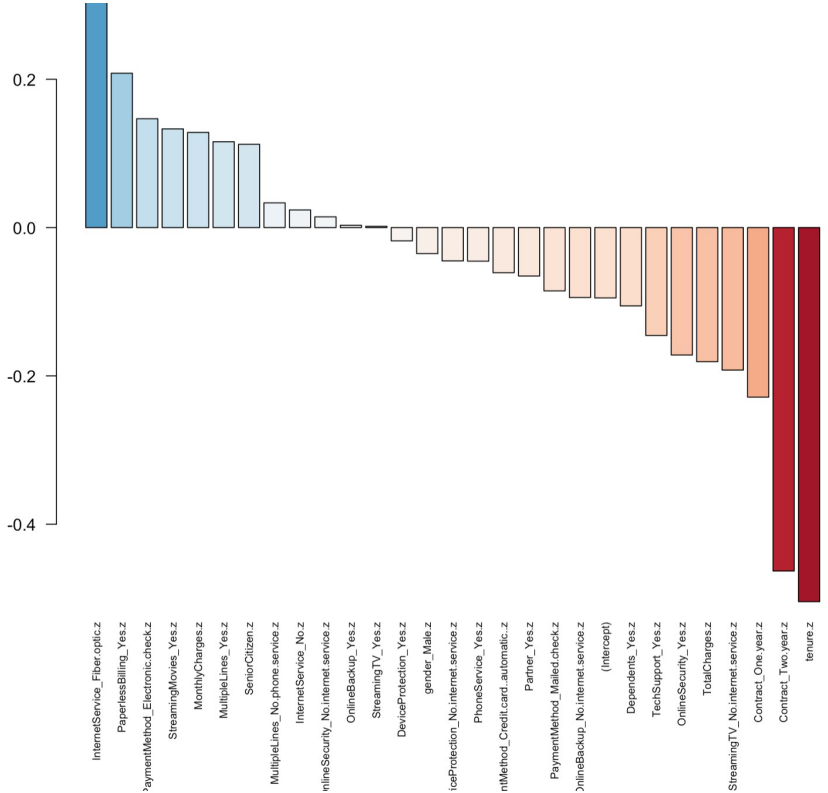
- For example, suppose we only have one feature: tenure
- Then our logistic regression is

$$\ln\left(\frac{P_{churned}}{1-p_{churned}}\right) = \beta_0 + \beta_1 \text{ (tenure)}$$

- After fitting for the coefficients, $\beta$'s, we get that $\beta_1$ = -0.91
  - In logistic regression, each exp($\beta$) is a measure of how much the odds change when we increase that $\beta$'s regressor by 1 unit
    - In our case, exp($\beta_1$) = exp(-0.91) = 0.4
    - This means that **each 1-month increase in tenure leads to a decrease in the odds of churn by a factor of 0.4**
      - In other words, each 1 month stay as a customer makes one 2.5 times likely to remain (1/0.4 = 2.5)

# Churn: Logistic Regression, all regressors

- We can now add more regressors or features to our logistic regression model
- All of them will be nicely interpretable in a similar fashion
- What's more, with regression you also get the signs of the coefficients
  - The sign also easily interpretable - is the variable "helping" or "hurting" what you are trying to predict

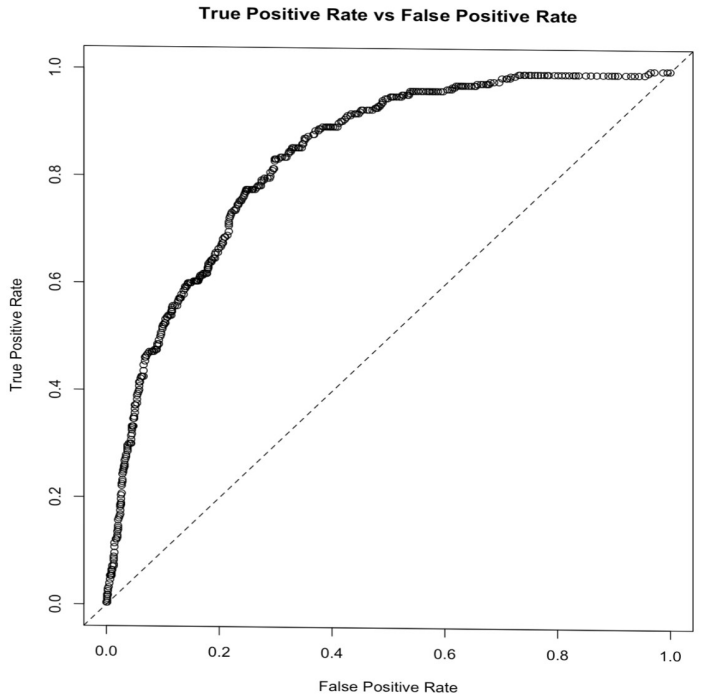


# Churn Logistic Regression: Performance

- The model is definitely easy to interpret
  - But here's the $60M question: how well is it performing?
  - Let's use the "area under the curve" metric
    - The closer to 1, the better
    - For this model, the area = 0.80
  - Other metrics: confusion matrix

| | 0 | 1 | Error | Rate |
|---|---|---|---|---|
| 0 | 582 | 192 | 0.248062 | =192/774 |
| 1 | 63 | 217 | 0.225000 | =63/280 |
| Totals | 645 | 409 | 0.241935 | =255/1054 |

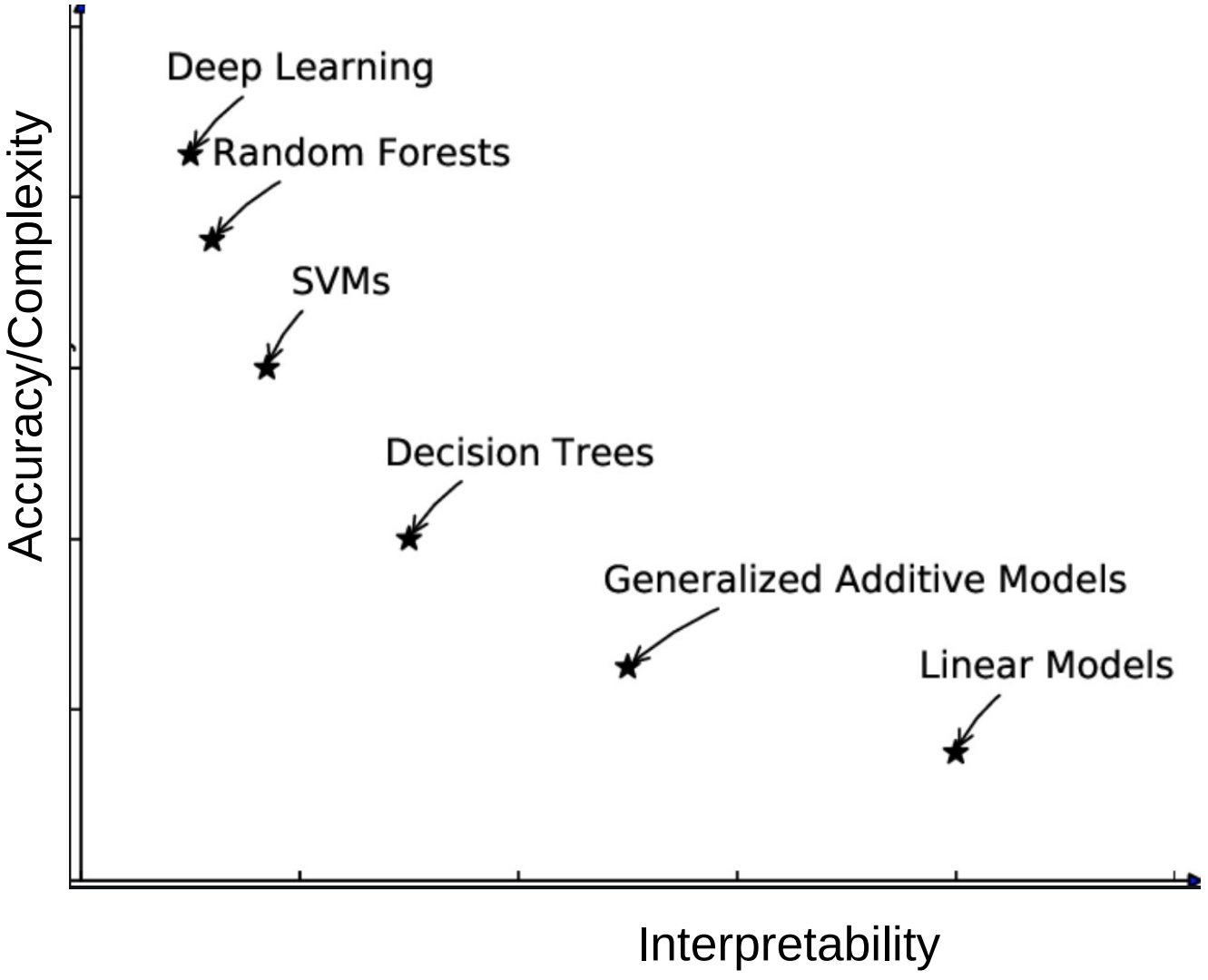
**We might do better with a more complex model**



# Interpretability vs. Predictive Power trade-off

- While easily interpretable, many simple models are *too* simple for many real world phenomena
  - For example, image and speech recognition require that the input – output map be insensitive to things like variations in positions, orientation, or illumination
    - e.g., two pictures of the same dog in different poses
  - But it has to be sensitive to “local” patterns
    - e.g., the dog's anatomy vs. that of a similar-posed wolf
  - You can’t achieve both of these with simple models

- Typically, the more complex models become more challenging to interpret

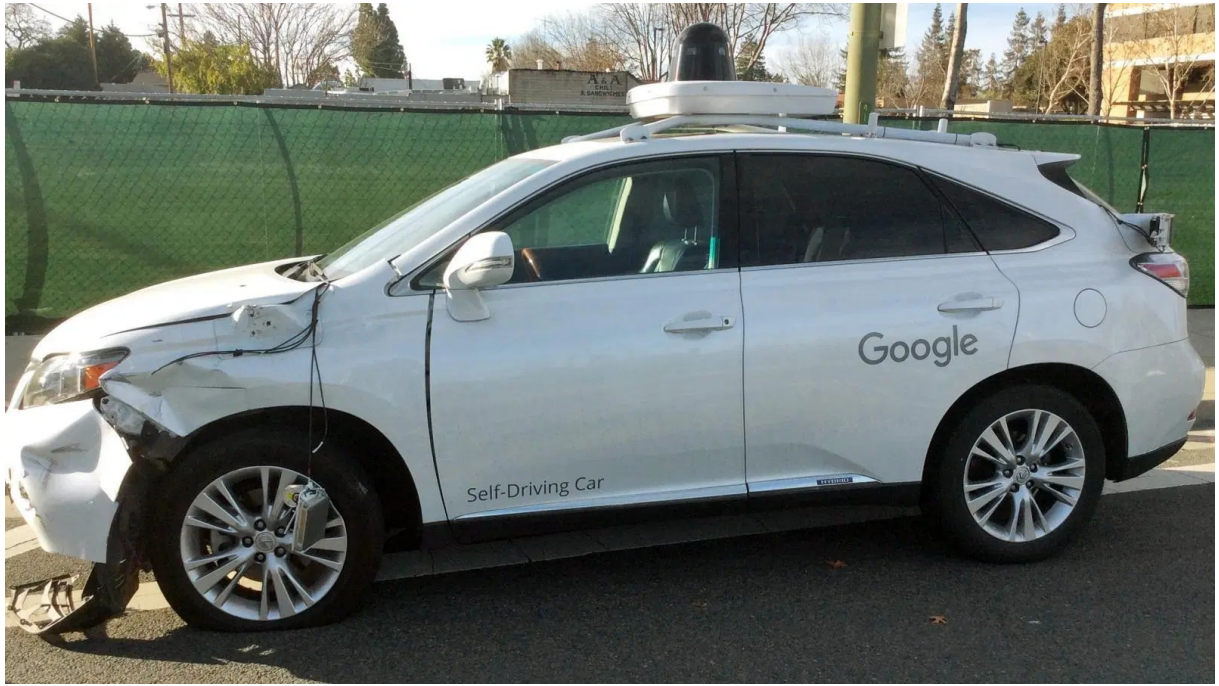

Deep Learning
Random Forests
SVMs
Decision Trees
Generalized Additive Models
Linear Models
Accuracy/Complexity
Interpretability

# Why does interpretability even matter?

- One might argue that, as long as a model results in great performance, it does not matter how it did it

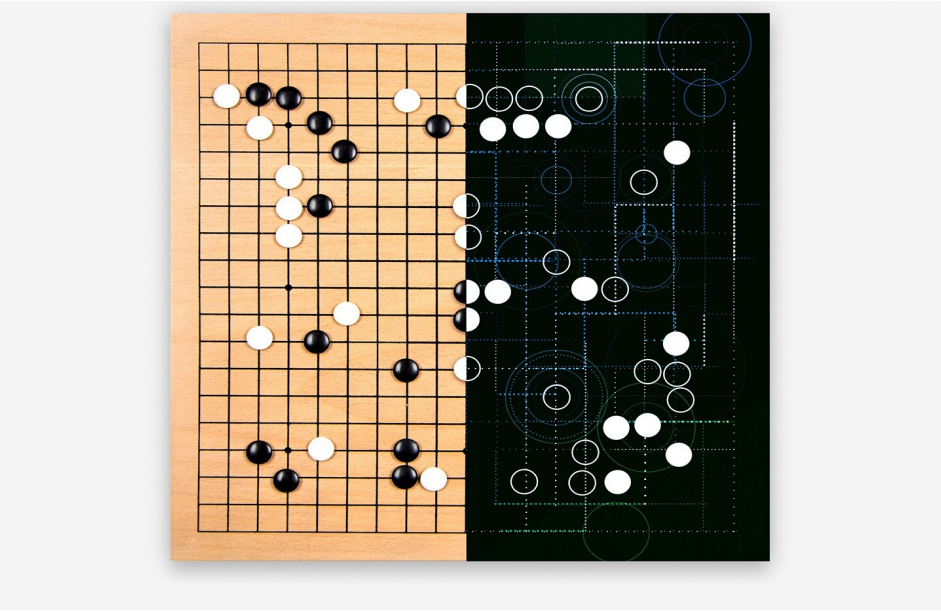
- But that's short-sighted, because...

# Model’s wrong decisions can be dangerous



Models can be improved after human inspection

Interpreting complex models can help us understand our own brain better (e.g., would a human ever make a move like this?)



Some model outputs are useless unless interpreted in a specific way (e.g., “there’s a 59% chance a patient will develop cancer in the next year” - what does a doctor do with this type of information?)

# Compliance with new legislature (e.g., GDRP)

# Churn: More Complex Model

- Our churn data might not be the best candidate for more complex treatments
  - Deep learning, in particular, really shines on complex, unstructured data such as speech, images, etc.

- But we will use it anyway because we are already familiar with the data and the simpler model

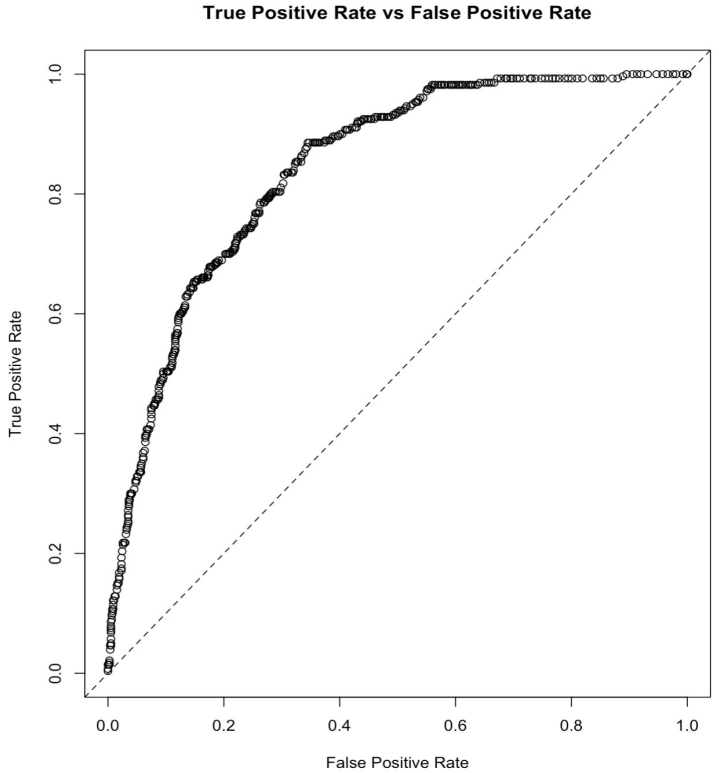
# Churn: More Complex Models

- An automatically selected by the H20 package “GBM” (**G**radient **B**oosting **M**achine) is a much more complex, non-linear model
  - It resulted in an area under the curve = 0.85 on the test data
  - Confusion Matrix:

| | 0 | 1 | Error | Rate |
|---|---|---|---|---|
| 0 | 659 | 115 | 0.148579 | =115/774 |
| 1 | 97 | 183 | 0.346429 | =97/280 |
| Totals | 756 | 298 | 0.201139 | =212/1054 |

  - Not much of an improvement, but still!

- We want to use this better-performing model, but do we trust it?



# Model Interpretability: High-Level Overview

There are three main approaches to interpreting complex ML models:

- Model-agnostic
  - Works for any deep learning as well as for “traditional” machine learning models
- Model-specific
  - Leverages a specific model’s internal structure to gain insights specific to this model
- Out of the box interpretable models
  - Models that are designed to be inherently interpretable

# Model Interpretability: High-Level Overview

There are three main approaches to interpreting complex ML models:

- **Model-agnostic**
  - Works for any deep learning as well as for “traditional” machine learning models
- Model-specific
  - Leverages a specific model’s internal structure to gain insights specific to this model
- Out of the box interpretable models
  - Models that are designed to be inherently interpretable

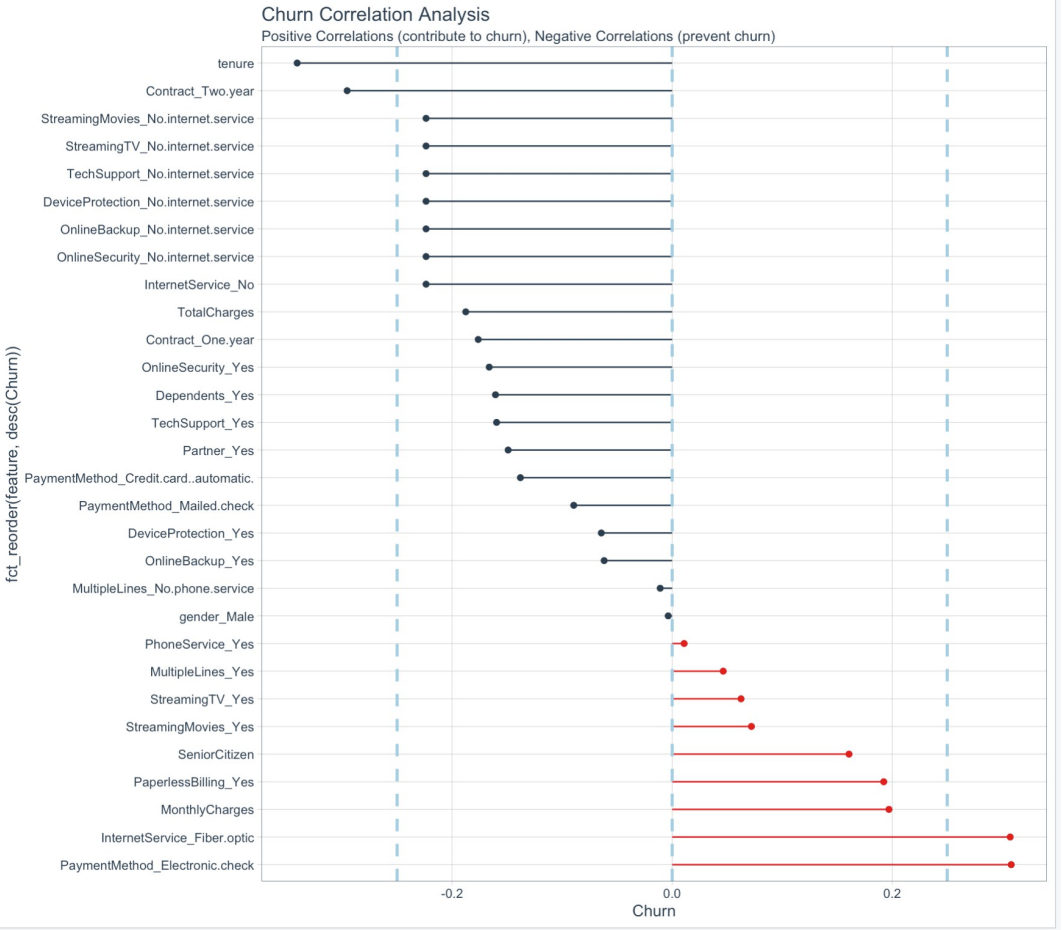
# Model-Agnostic Interpretability

These approaches work for any model regardless of complexity

- Global:
  - Correlation analysis
  - Partial dependence plots for each feature = marginal effects of that feature on model predictions
- Local:
  - LIME (Local Interpretable Model-agnostic Explanations)
  - SHAP (Shapley values)

# Global Approach: Correlation Analysis

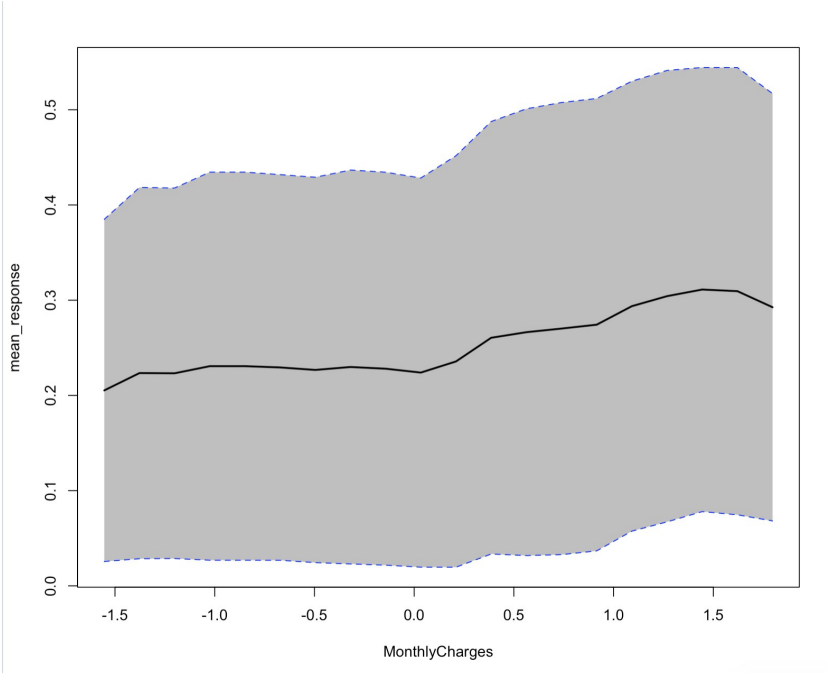
- Correlation analysis is a simple assessment of which variables are negatively or positively correlated with variable you want to predict (Churn)
- It's not predictive, but is informative



# Global Approach: Partial Dependence Plots

- Partial dependence plots can help us glean insights into where a given feature contributes the most to the prediction

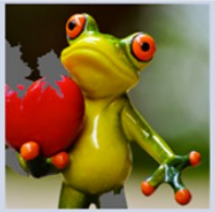
- For example, feature “monthly charges” seems to promote churn in an almost linear fashion after a certain point

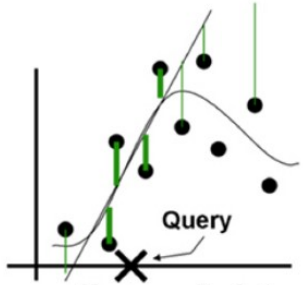


# Local model interpretability: LIME

- LIME = **L**ocal **I**nterpretable **M**odel-Agnostic **E**xplanations
- Main assumption: every complex model is linear on a local scale
  - “Fit globally, explain locally”
- How LIME works:
  - Take one instance of your data (e.g., one record of a customer, one image, etc.)
  - Change it in some way (e.g., obscure parts of the image)
  - Get a prediction from the model for that changed instance
  - Learn a simple (linear) model on a bunch of such changed instances, taking care to account for how similar the changes instances are to the original one
  - This way, we learn which features of the original instance are more important than others

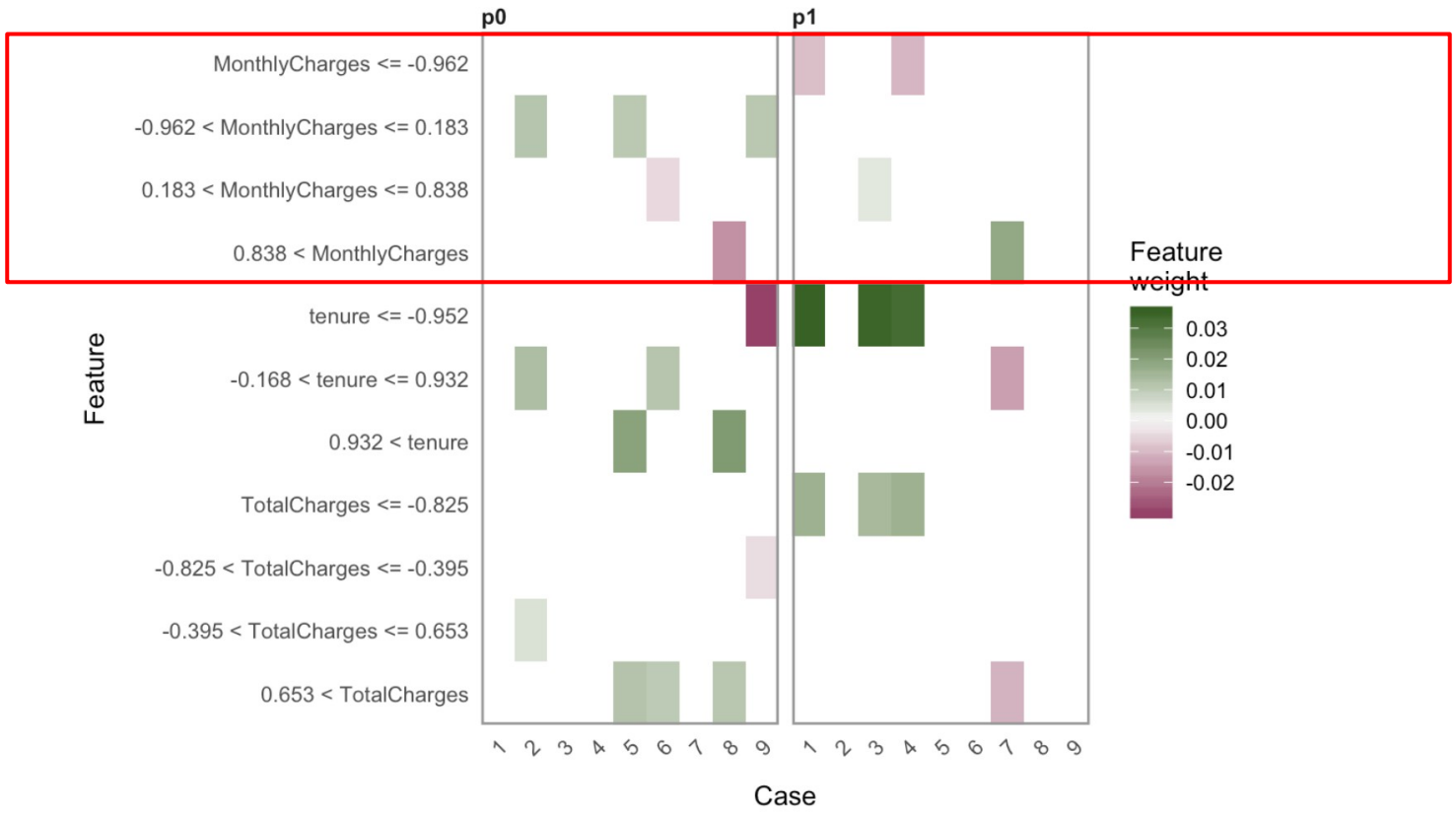
Original Image
P(tree frog) = 0.54

| Perturbed Instances | P(tree frog) |
| --- | --- |
|  | 0.85 |
|  | 0.00001 |
|  | 0.52 |



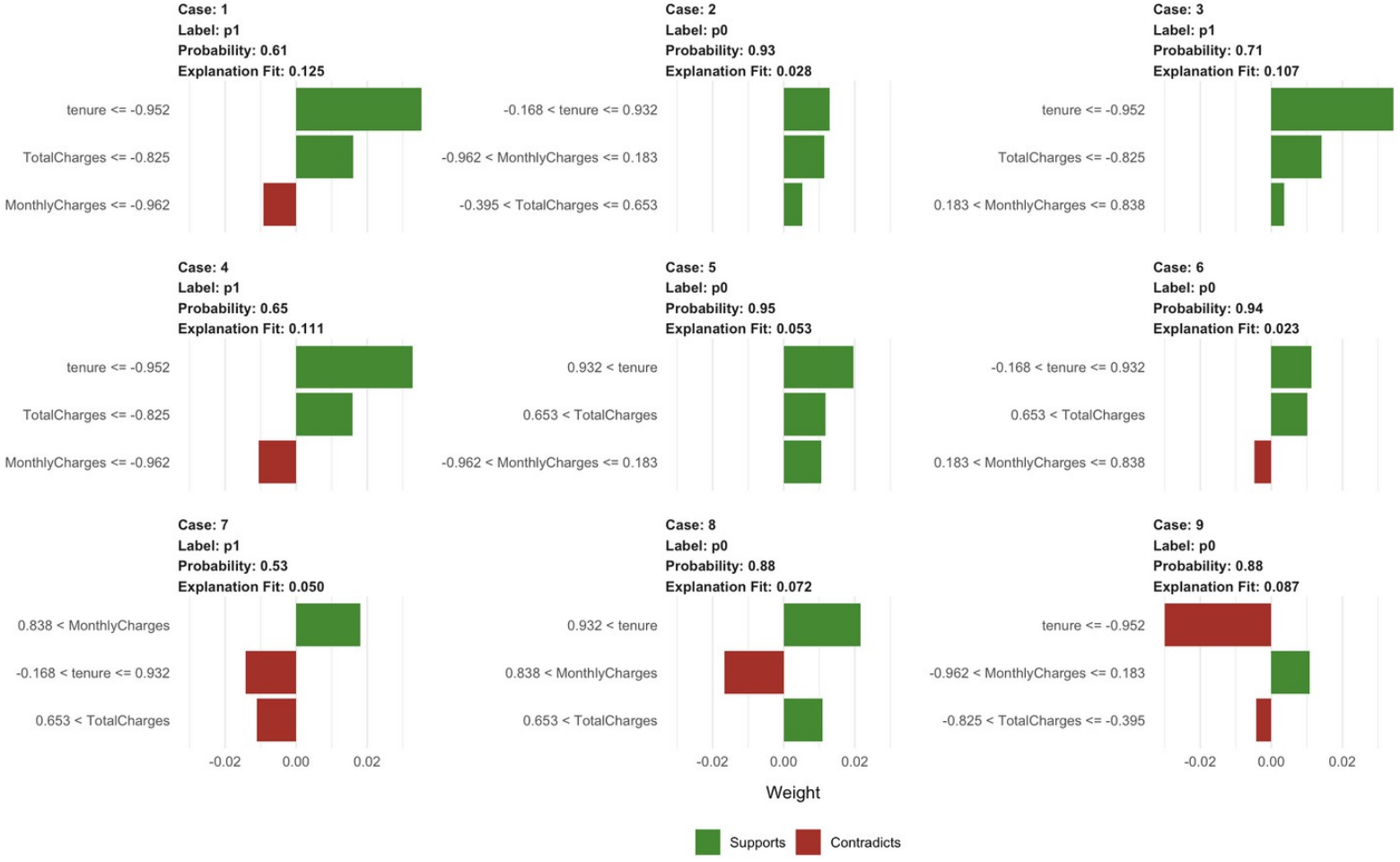

Locally weighted regression

Explanation


p0
p1
MonthlyCharges <= -0.962
-0.962 < MonthlyCharges <= 0.183
0.183 < MonthlyCharges <= 0.838
0.838 < MonthlyCharges
tenure <= -0.952
-0.168 < tenure <= 0.932
0.932 < tenure
TotalCharges <= -0.825
-0.825 < TotalCharges <= -0.395
-0.395 < TotalCharges <= 0.653
0.653 < TotalCharges
Feature
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
Case
Feature weight
0.03
0.02
0.01
0.00
-0.01
-0.02

# How do we interpret this plot?


- For each of our 10 sample cases, we get a “personalized” feature breakdown
  - Consider variable “monthly charges”:
    - Cases 2, 5, and 9 were labeled non-churners; this variable being on the smaller side supported that designation
    - Case 7 was labeled a churner; this variable being larger supported that designation

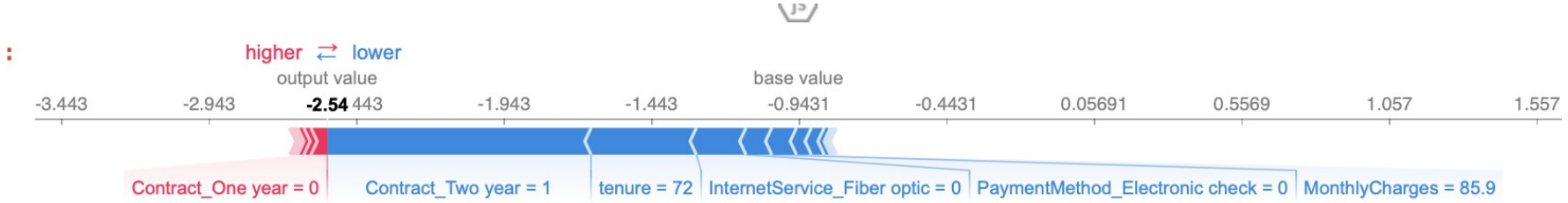

Case: 1
Label: p1
Probability: 0.61
Explanation Fit: 0.125
tenure <= -0.952
TotalCharges <= -0.825
MonthlyCharges <= -0.962
Case: 2
Label: p0
Probability: 0.93
Explanation Fit: 0.028
-0.168 < tenure <= 0.932
-0.962 < MonthlyCharges <= 0.183
-0.395 < TotalCharges <= 0.653
Case: 3
Label: p1
Probability: 0.71
Explanation Fit: 0.107
tenure <= -0.952
TotalCharges <= -0.825
0.183 < MonthlyCharges <= 0.838
Case: 4
Label: p1
Probability: 0.65
Explanation Fit: 0.111
tenure <= -0.952
TotalCharges <= -0.825
MonthlyCharges <= -0.962
Case: 5
Label: p0
Probability: 0.95
Explanation Fit: 0.053
0.932 < tenure
0.653 < TotalCharges
-0.962 < MonthlyCharges <= 0.183
Case: 6
Label: p0
Probability: 0.94
Explanation Fit: 0.023
-0.168 < tenure <= 0.932
0.653 < TotalCharges
0.183 < MonthlyCharges <= 0.838
Case: 7
Label: p1
Probability: 0.53
Explanation Fit: 0.050
0.838 < MonthlyCharges
-0.168 < tenure <= 0.932
0.653 < TotalCharges
-0.02
0.00
0.02
Case: 8
Label: p0
Probability: 0.88
Explanation Fit: 0.072
0.932 < tenure
0.838 < MonthlyCharges
0.653 < TotalCharges
-0.02
0.00
0.02
Case: 9
Label: p0
Probability: 0.88
Explanation Fit: 0.087
tenure <= -0.952
-0.962 < MonthlyCharges <= 0.183
-0.825 < TotalCharges <= -0.395
-0.02
0.00
0.02
Weight
Supports
Contradicts

# Local model interpretability: SHAP

- SHAP = **SH**apley **A**dditive Ex**P**lanation, is an approach borrowed from game theory
  - In a coalition with cooperative multiple players with different contributions, what's the fairest way to divide up the eventual payoff?

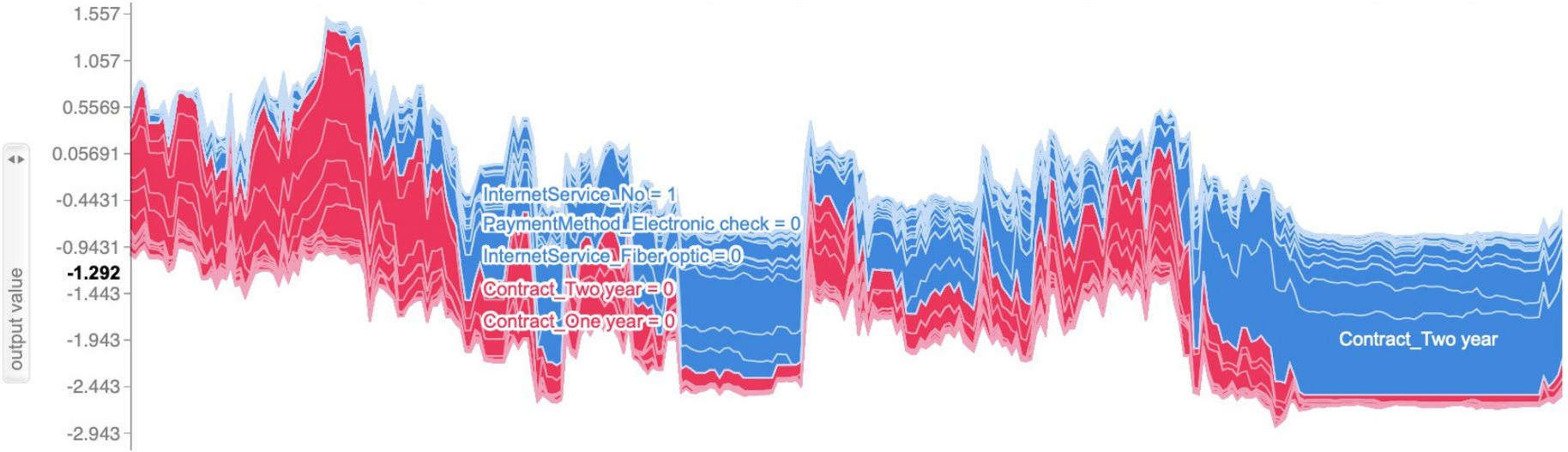
- Basic idea: for each individual instance (one customer with their early usage features), the features "play a game" together, in which the prediction (churn) is the payoff
  - The Shapley values tell us the contributions of each feature to the prediction of this particular instance compared to the average prediction for the dataset

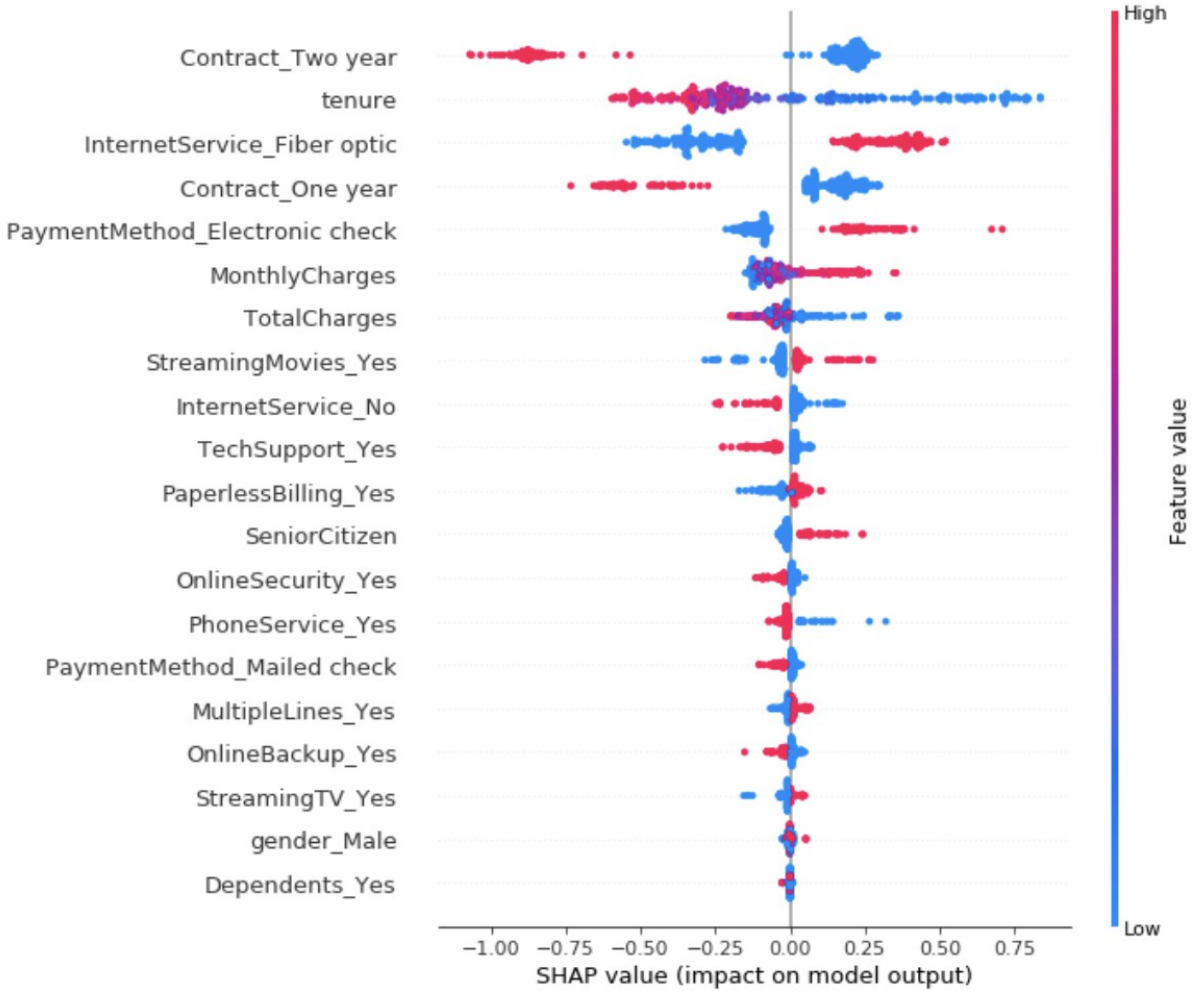
# SHAP for Churn: one particular instance




- This is one way to show how different features push the model output one way or another for this particular customer
- Features pushing the prediction higher (towards churn) are shown in red, those pushing the prediction lower (towards no churn) are in blue

# SHAP for Churn: whole training sample




Contract_Two year
tenure
InternetService_Fiber optic
Contract_One year
PaymentMethod_Electronic check
MonthlyCharges
TotalCharges
StreamingMovies_Yes
InternetService_No
TechSupport_Yes
PaperlessBilling_Yes
SeniorCitizen
OnlineSecurity_Yes
PhoneService_Yes
PaymentMethod_Mailed check
MultipleLines_Yes
OnlineBackup_Yes
StreamingTV_Yes
gender_Male
Dependents_Yes
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75
SHAP value (impact on model output)
High
Feature value
Low

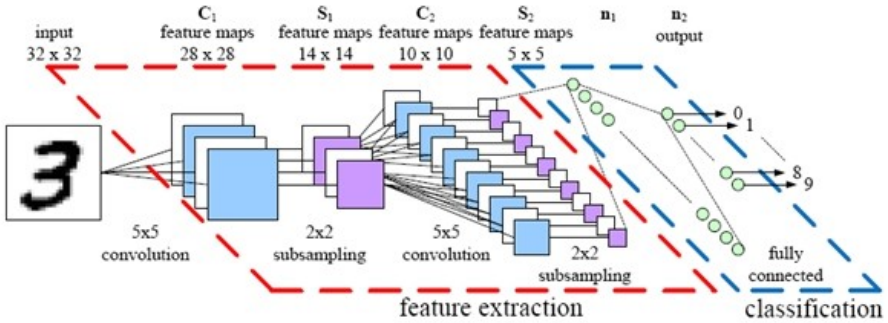
# Model Interpretability: High-Level Overview

There are three main approaches to interpreting complex ML models:

- Model-agnostic
  - Works for any deep learning as well as  “traditional” machine learning models
- **Model-specific**
  - Leverages a specific model’s internal structure to gain insights specific to this model
- Out of the box interpretable models
  - Models that are designed to be inherently interpretable

# Model-specific interpretability: Convolutional Neural Networks

- Consider what looks like a black box model: convolutional neural networks (CNNs) for image processing




- We can ask several questions:
  - Is there any way to peek inside the model to see what it's actually doing?
  - Is there a way to gain more trust in the results by highlighting relevant image aspects that contributed to a prediction?
    - Saliency maps
    - Class activation maps
    - Text-based explanations

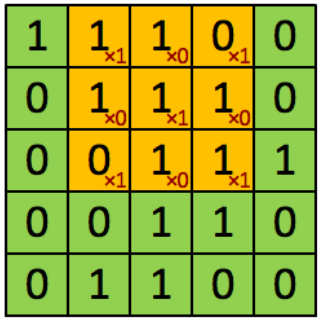
# Model-specific interpretability: Peeking Inside CNNs

- In order to gain some insights into what CNNs are actually doing, we can visualize what’s happening to our inputs at every step of the CNN pipeline

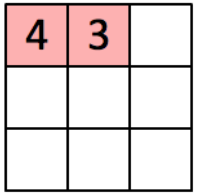
- A typical pipeline has the following steps:
  - Convolution
  - Pooling
  - Activation

# Visualizing Convolution

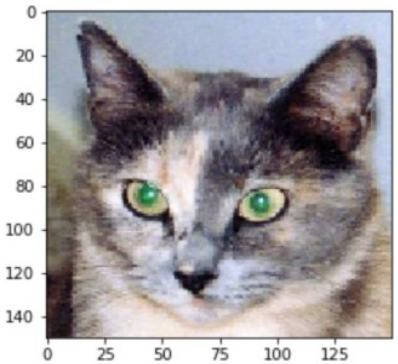
- Convolution is essentially filtering
  - Convolutional layers detect local conjunctions of features
- For example:



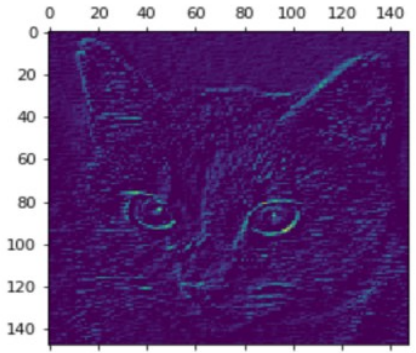

Image



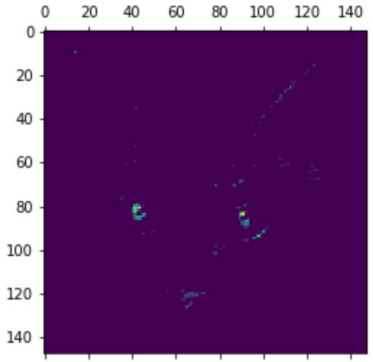

Convolved Feature




Original image



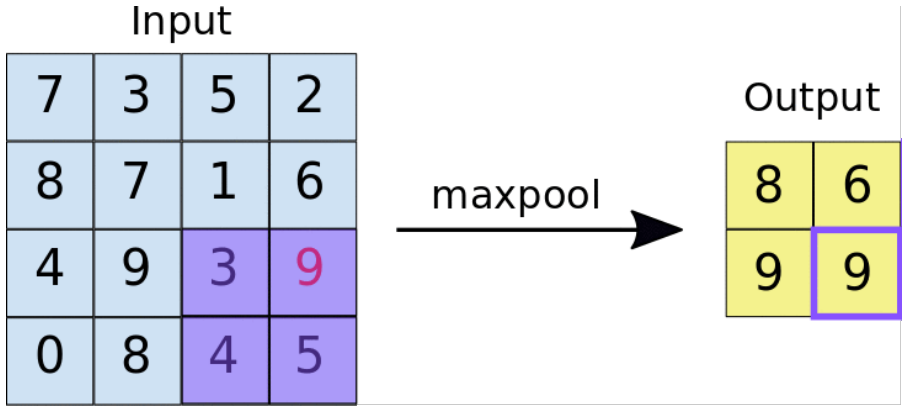

Diagonal Edge Detector




“Bright Green Dot” detector

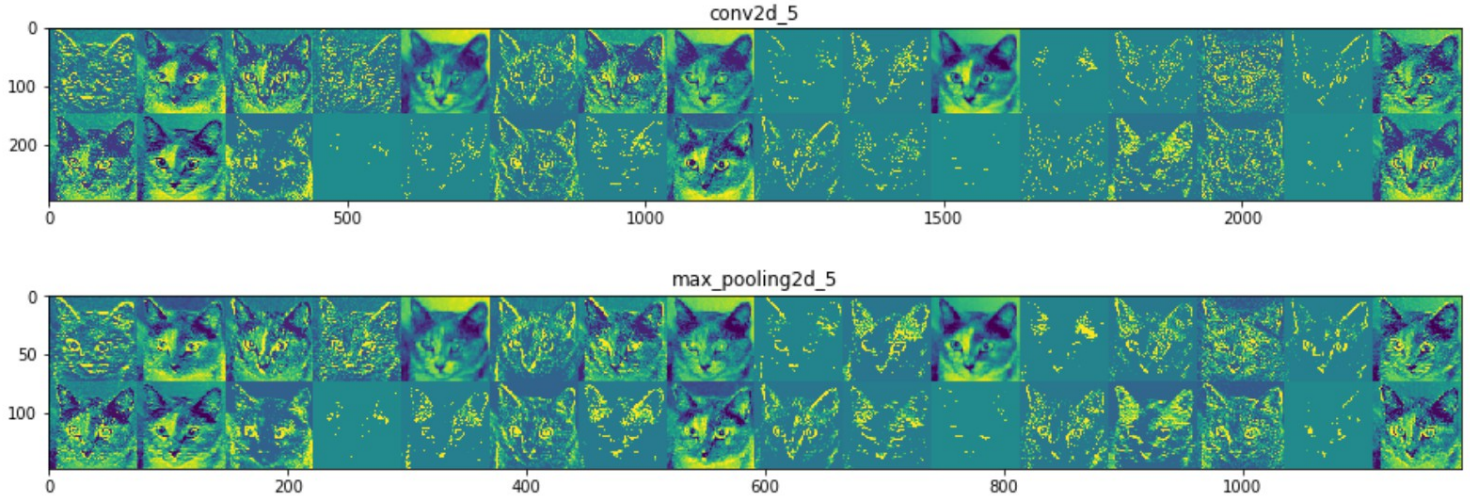
# Visualizing Pooling

- Pooling is a layer in an image processing network that reduces the feature dimensionality
  - Can be max, min, average, …, pooling of a collection of pixels




- Pooling layers' role is to merge semantically similar features into one

# Intermediate results of convolution and pooling

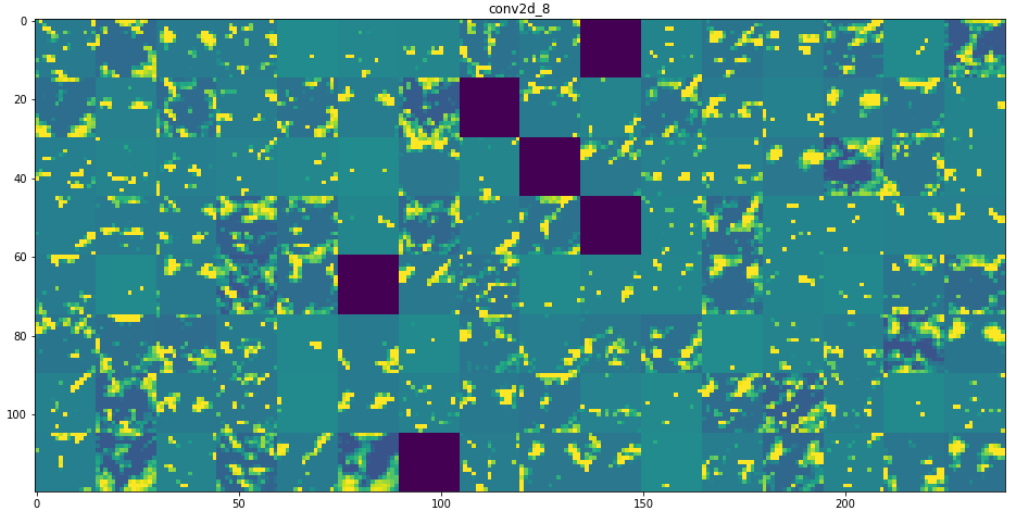


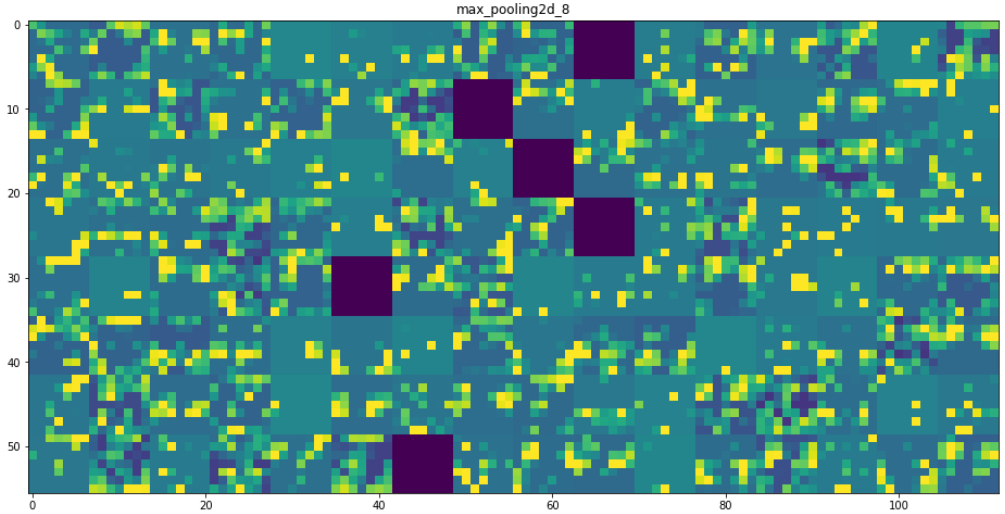

(Note the reduced dimensionality of the images after the pooling step)

# This method of “peeking” into CNNs has its limits

- As we go deeper into the layers of our deep neural network, the images become increasingly abstract and less visually obvious (see next page)

- This is because they start to encode higher-level concepts
  - So less info about the visual content of the image, more info related to the class of the image (e.g., “cat” or “dog”)


conv2d_8
0
20
40
60
80
100
0
50
100
150
200

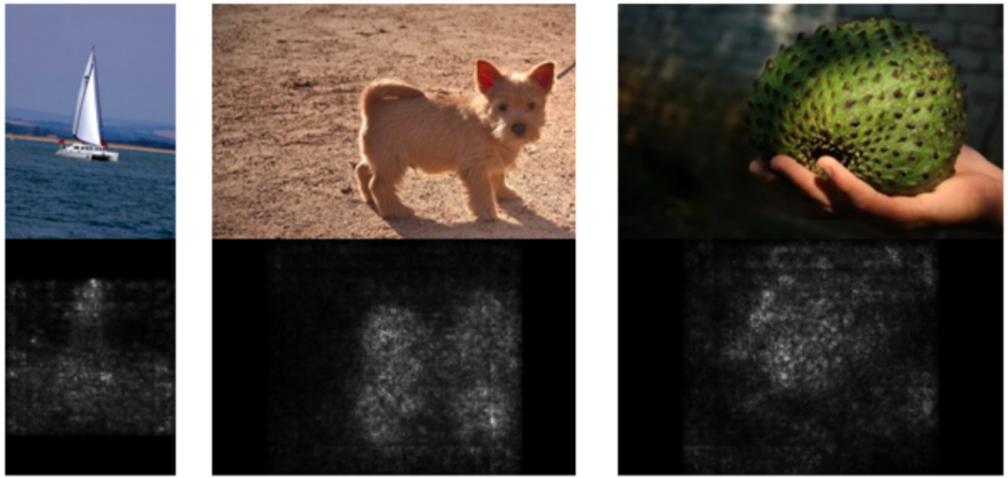

max_pooling2d_8
0
10
20
30
40
50
0
20
40
60
80
100

# Gaining trust in CNN predictions: Class Activation Maps

- **Class Activation Maps (CAMs)** are heatmaps that are essentially scores associated with a specific prediction, computed for every location in the input images
- This allows us to judge how important each location is with respect to the eventual prediction

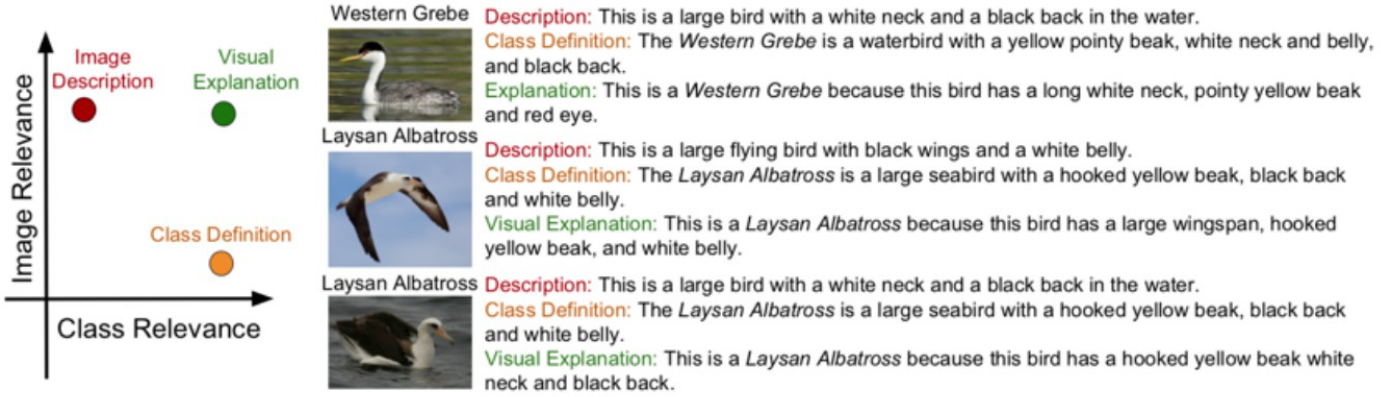
# Gaining trust in CNN predictions: Saliency Maps

- **Saliency Maps** highlight the relevant input pixels to which the output score is particularly sensitive



# Gaining trust in CNN predictions: text-based explanations

- The idea is to provide not only a classification of an image, but also some text-based justification for the classification
  - The text should contain both a description of the image, but also the aspects of the image that contributed to its classification as an instance of a particular class



# Model Interpretability: High-Level Overview

There are three main approaches to interpreting complex ML models:

- Model-agnostic
  - Works for  any deep learning as well as for “traditional” machine learning models
- Model-specific
  - Leverages a specific model’s internal structure to gain insights specific to this model
- **Out of the box interpretable models**
  - Models that are designed to be inherently interpretable

# Out-of-the-box Interpretable Models

- Here, one aims to develop inherently interpretable models right from the start
- Typically, the models are decision sets, decision trees, linear models…
- Consider decision sets as an example:
  - Rule-based, where the optimal set of rules is found from a candidate rule pool
  - Aims to maximize both accuracy and interpretability

# Conclusion

- There appears to be an inherent trade-off in complexity vs. interpretability of machine learning models
  - Deep learning models are a prime example of this
- Maintaining interpretability is crucial
- There are a number of approaches for making complex models more “human-friendly”
  - Model-agnostic
  - Model-specific
  - Inherently interpretable
- In this talk, we only saw a sliver of possible approaches to interpretability
- There’s no doubt that focus on interpretability will remain as a) data science models become ever more complex, and b) demand for transparency increases