Interpretable Machine Learning with Python

Learn to build interpretable high-performance models with hands-on real-world examples

Serg Masís

Interpretable Machine Learning with Python

Learn to build interpretable high-performance models with hands-on real-world examples

Serg Masís

BIRMINGHAM—MUMBAI

Interpretable Machine Learning with Python

Group Product Manager: Kunal Parikh
Publishing Product Manager: Sunith Shetty
Acquisition Editor: Reshma Raman
Senior Editor: Roshan Kumar
Content Development Editors: Sean Lobo and Joseph Sunil
Technical Editor: Sonam Pandey
Copy Editor: Safis Editing
Project Coordinator: Aishwarya Mohan
Proofreader: Safis Editing
Indexer: Priyanka Dhadke
Production Designer: Roshan Kawale

First published: March 2021
Production reference: 1250321

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-80020-390-7

www.packt.com

1

Interpretation, Interpretability, and Explainability; and Why Does It All Matter?

We live in a world whose rules and procedures are governed by data and algorithms.

For instance, there are rules as to who gets approved for credit or released on bail, and which social media posts might get censored. There are also procedures to determine which marketing tactics are most effective and which chest x-ray features might diagnose a positive case of pneumonia.

You expect this because it is nothing new!

But not so long ago, rules and procedures such as these used to be hardcoded into software, textbooks, and paper forms, and humans were the ultimate decision-makers. Often, it was entirely up to human discretion. Decisions depended on human discretion because rules and procedures were rigid and, therefore, not always applicable. There were *always* exceptions, so a human was needed to make them.

For example, if you would ask for a mortgage, your approval depended on an acceptable and reasonably lengthy credit history. This data, in turn, would produce a credit score using a scoring algorithm. Then, the bank had rules that determined what score was good enough for the mortgage you wanted. Your loan officer could follow it or override it.

These days, financial institutions train models on thousands of mortgage outcomes, with dozens of variables. These models can be used to determine the likelihood that you would default on a mortgage with a presumed high accuracy. If there is a loan officer to stamp the approval or denial, it's no longer merely a guideline but an algorithmic decision. How could it be wrong? How could it be right?

Hold on to that thought because, throughout this book, we will be learning the answers to these questions and many more!

To interpret decisions made by a machine learning model is to find meaning in it, but furthermore, you can trace it back to its source and the process that transformed it. This chapter introduces machine learning interpretation and related concepts such as interpretability, explainability, black-box models, and transparency. This chapter provides definitions for these terms to avoid ambiguity and underpins the value of machine learning interpretability. These are the main topics we are going to cover:

- What is machine learning interpretation?
- Understanding the difference between interpretation and explainability
- A business case for interpretability

Let's get started!

Technical requirements

To follow the example in this chapter, you will need Python 3, either running in a Jupyter environment or in your favorite **integrated development environment** (**IDE**) such as PyCharm, Atom, VSCode, PyDev, or Idle. The example also requires the `requests`, `bs4`, `pandas`, `sklearn` , `matplotlib`, and `scipy` Python libraries. The code for this chapter is located here: `https://github.com/PacktPublishing/Interpretable-Machine-Learning-with-Python/tree/master/Chapter01`.

What is machine learning interpretation?

To interpret something is to *explain the meaning of it*. In the context of machine learning, that something is an algorithm. More specifically, that algorithm is a mathematical one that takes input data and produces an output, much like with any formula.

Let's examine the most basic of models, simple linear regression, illustrated in the following formula:

$$\hat{y} = \beta_0 + \beta_1 x_1$$

Once fitted to the data, the meaning of this model is that $\hat{y}$ predictions are a weighted sum of the x features with the β coefficients. In this case, there's only one x **feature** or **predictor** variable, and the y variable is typically called the **response** or **target** variable. A simple linear regression formula single-handedly explains the transformation, which is performed on the input data x_1 to produce the output $\hat{y}$. The following example can illustrate this concept in further detail.

Understanding a simple weight prediction model

If you go to this web page maintained by the University of California, `http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_Dinov_020108_HeightsWeights`, you can find a link to download a dataset of 25,000 synthetic records of weights and heights of 18-year-olds. We won't use the entire dataset but only the sample table on the web page itself with 200 records. We scrape the table from the web page and fit a linear regression model to the data. The model uses the height to predict the weight.

In other words, $x_1 = \text{height}$ and $y = \text{weight}$, so the formula for the linear regression model would be as follows:

$$\text{weight} = \beta_0 + \beta_1 \text{height}$$

You can find the code for this example here: `https://github.com/PacktPublishing/Interpretable-Machine-Learning-with-Python/blob/master/Chapter01/WeightPrediction.ipynb`.

To run this example, you need to install the following libraries:

- `requests` to fetch the web page
- `bs4` (Beautiful Soup) to scrape the table from the web page
- `pandas` to load the table in to a dataframe
- `sklearn` (scikit-learn) to fit the linear regression model and calculate its error

- `matplotlib` to visualize the model
- `scipy` to test the correlation

You should load all of them first, as follows:

```
Import math
import requests
from bs4 import BeautifulSoup
import pandas as pd
from sklearn import linear_model
from sklearn.metrics import mean_absolute_error
import matplotlib.pyplot as plt
from scipy.stats import pearsonr
```

Once the libraries are all loaded, you use `requests` to fetch the contents of the web page, like this:

```
url = \
'http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_
Dinov_020108_HeightsWeights'
page = requests.get(url)
```

Then, take these contents and scrape out just the contents of the table with `BeautifulSoup`, as follows:

```
soup = BeautifulSoup(page.content, 'html.parser')
tbl = soup.find("table",{"class":"wikitable"})
```

`pandas` can turn the raw **HyperText Markup Language** (**HTML**) contents of the table into a dataframe, as illustrated here:

```
height_weight_df = pd.read_html(str(tbl))[0]\
[['Height(Inches)','Weight(Pounds)']]
```

And voilà! We now have a dataframe with `Heights(Inches)` in one column and `Weights(Pounds)` in another. As a sanity check, we can then count the number of records. This should be 200. The code is shown here:

```
num_records = height_weight_df.shape[0]
print(num_records)
```

Now that we have confirmed that we have the data, we must transform it so that it conforms to the model's specifications. `sklearn` needs it as NumPy arrays with (200,1) dimensions, so we must first extract the `Height(Inches)` and `Weight(Pounds)` pandas Series. Then, we turn them into (200,) NumPy arrays, and, finally, reshape them into (200,1) dimensions. The following commands perform all the necessary transformation operations:

```
x = height_weight_df['Height(Inches)'].values.\
                                        reshape(num_records, 1)
y = height_weight_df['Weight(Pounds)'].values.\
                                        reshape(num_records, 1)
```

Then, you initialize the scikit-learn `LinearRegression` model and `fit` it with the training data, as follows:

```
model = linear_model.LinearRegression()
_ = model.fit(x,y)
```

To output the fitted linear regression model formula in scikit-learn, you must extract the intercept and coefficients. This is the **formula** that explains how it makes predictions:

```
print("ŷ =" + str(model.intercept_[0]) + " + " +\
                            str(model.coef_.T[0][0]) + " x1")
```

The following is the output:

```
ŷ = -106.02770644878132 + 3.432676129271629 x1
```

This tells us that, on average, for every additional pound, there are 3.4 inches of height.

However, *explaining how the model works* is only one way to explain this linear regression model, and this is only one side of the story. The model isn't perfect because the actual outcomes and the predicted outcomes are not the same for the training data. The difference between both is the **error** or **residuals**.

There are many ways of understanding an error in a model. You can use an error function such as `mean_absolute_error` to measure the deviation between the predicted values and the actual values, as illustrated in the following code snippet:

```
y_pred = model.predict(x)
mae = mean_absolute_error(y, y_pred)
print(mae)
```

The following is the output:

```
7.7587373803882205
```

A 7.8 mean absolute error means that, on average, the prediction is 7.8 pounds from the actual amount, but this might not be intuitive or informative. Visualizing the linear regression model can shed some light on how accurate these predictions truly are.

This can be done by using a `matplotlib` scatterplot and overlaying the linear model (in blue) and the *mean absolute error* (as two parallel bands in gray), as shown in the following code snippet:

```
plt.scatter(x, y, color='black')
plt.plot(x, y_pred, color='blue', linewidth=3)
plt.plot(x, y_pred + mae, color='lightgray')
plt.plot(x, y_pred - mae, color='lightgray')
plt.xlabel('Height(Inches)')
plt.ylabel('Weight(Pounds)')
```

If you run the preceding snippet, the plot shown here in *Figure 1.1* is what you get as the output:

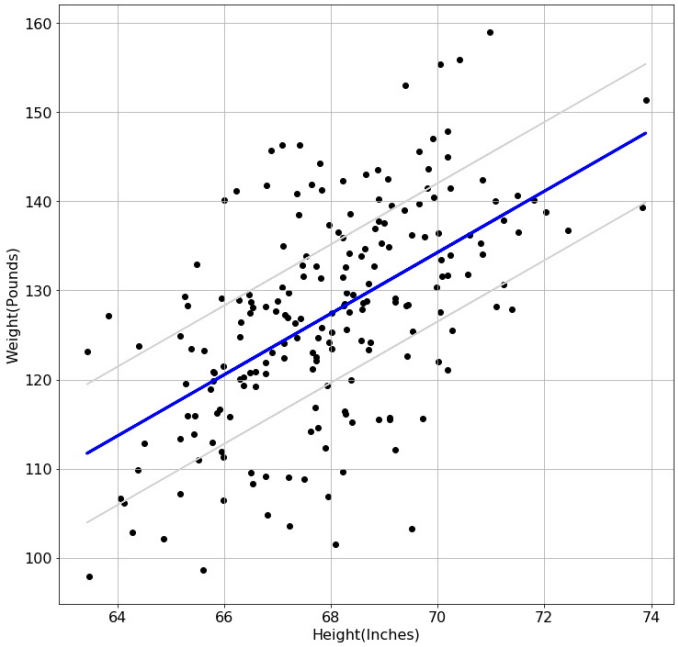

Figure 1.1 – Linear regression model to predict weight based on height

As you can appreciate from the plot in *Figure 1.1*, there are many times in which the actuals are $20 - 25$ pounds away from the prediction. Yet the mean absolute error can fool you into thinking that the error is always closer to 8. This is why it is essential to visualize the error of the model to understand its distribution. Judging from this graph, we can tell that there are no red flags that stand out about this distribution, such as residuals being more spread out for one range of heights than for others. Since it is more or less equally spread out, we say it's **homoscedastic**. In the case of linear regression, this is one of many model assumptions you should test for, along with *linearity*, *normality*, *independence*, and lack of *multicollinearity* (if there's more than one feature). These assumptions ensure that you are using the right model for the job. In other words, the height and weight *can be explained* with a linear relationship, and it is a good idea to do so, statistically speaking.

With this model, we are trying to establish a linear relationship between x height and y weight. This association is called a **linear correlation**. One way to measure this relationship's strength is with **Pearson's correlation coefficient**. This statistical method measures the association between two variables using their covariance divided by their standard deviations. It is a number between -1 and 1 whereby the closer the number it is to zero, the weaker the association is. If the number is positive, there is a positive association, and if it's negative, there is a negative one. In Python, you can compute Pearson's correlation coefficient with the `pearsonr` function from `scipy`, as illustrated here:

```
corr, pval = pearsonr(x[:,0], y[:,0])
print(corr)
```

The following is the output:

```
0.5568647346122992
```

The number is positive, which is no surprise because as height increases, weight also tends to increase, but it is also closer to 1 than to 0, denoting that it is strongly correlated. The second number produced by the `pearsonr` function is the p-value for testing non-correlation. If we test that it's less than an error level of 5%, we can say there's sufficient evidence of this correlation, as illustrated here:

```
print(pval < 0.05)
```

The following is the output:

```
True
```

Understanding how a model performs and in which circumstances can help us **explain why it makes certain predictions**, and when it cannot. Let's imagine we are asked to explain why someone who is 71 inches tall was predicted to have a weight of 134 pounds but instead weighed 18 pounds more. Judging from what we know about the model, this margin of error is not unusual even though it's not ideal. However, there are many circumstances in which we cannot expect this model to be reliable. What if we were asked to predict the weight of a person who is 56 inches tall with the help of this model? Could we assure the same level of accuracy? Definitely not, because we fit the model on the data of subjects no shorter than 63 inches. Ditto if we were asked to predict the weight of a 9-year-old, because the training data was for 18-year-olds.

Despite the acceptable results, this weight prediction model was not a realistic example. If you wanted to be more accurate but—more importantly—faithful to what can really impact the weight of an individual, you would need to add more variables. You can add—say—gender, age, diet, and activity level. This is where it gets interesting because you have to make sure **it is fair to include them, or not to include them**. For instance, if gender were included yet most of our dataset was composed of males, how could you ensure accuracy for females? This is what is called **selection bias**. And what if weight had more to do with lifestyle choices and circumstances such as poverty and pregnancy than gender? If these variables aren't included, this is called **omitted variable bias**. And then, does it make sense to include the sensitive gender variable at the risk of adding bias to the model?

Once you have multiple features that you have vetted for fairness, you can find out and *explain which features impact model performance*. We call this **feature importance**. However, as we add more variables, we increase the complexity of the model. Paradoxically, this is a problem for interpretation, and we will explore this in further detail in the following chapters. For now, the key takeaway should be that model interpretation has a lot to do with explaining the following:

1. Can we explain that predictions were made fairly?
2. Can we trace the predictions reliably back to something or someone?
3. Can we explain how predictions were made? Can we explain how the model works?

And ultimately, the question we are trying to answer is this:

Can we trust the model?

The three main concepts of interpretable machine learning directly relate to the three preceding questions and have the acronym of **FAT**, which stands for **fairness**, **accountability**, and **transparency**. If you can explain that predictions were made without discernible bias, then there is **fairness**. If you can explain why it makes certain predictions, then there's **accountability**. And if you can explain how predictions were made and how the model works, then there's **transparency**. There are many ethical concerns associated to these concepts, as shown here in *Figure 1.2*:

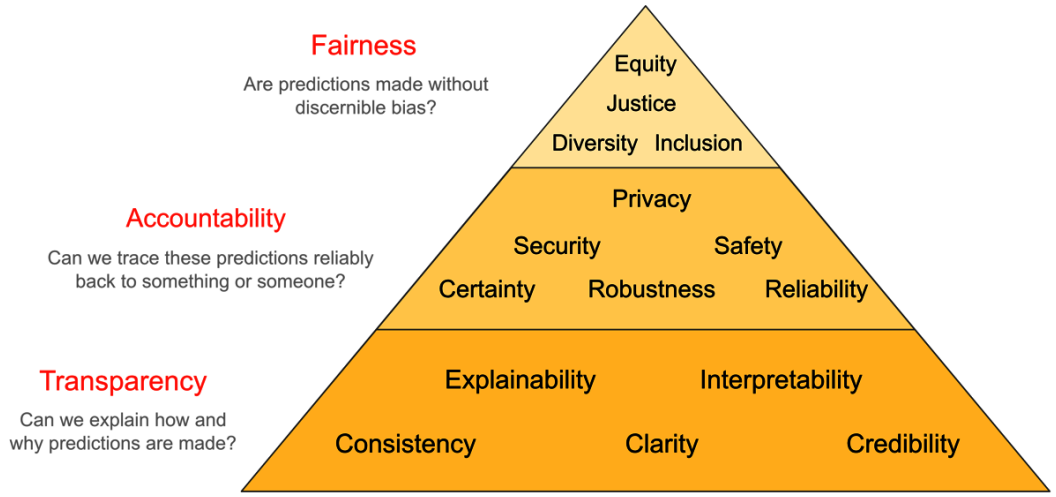

Figure 1.2 – Three main concept of Interpretable Machine Learning

Some researchers and companies have expanded FAT under a larger umbrella of ethical **artificial intelligence** (**AI**), thus turning FAT into FATE. Ethical AI is part of an even larger discussion of algorithmic and data governance. However, both concepts very much overlap since interpretable machine learning is how FAT principles and ethical concerns get implemented in machine learning. In this book, we will discuss ethics in this context. For instance, *Chapter 13*, *Adversarial Robustness* relates to reliability, safety, and security. *Chapter 11*, *Mitigating Bias and Causal Inference Methods* relates to fairness. That being said, interpretable machine learning can be leveraged with no ethical aim in mind, and also for unethical reasons.

Understanding the difference between interpretability and explainability

Something you've probably noticed when reading the first few pages of this book is that the verbs *interpret* and *explain*, as well as the nouns *interpretation* and *explanation*, have been used interchangeably. This is not surprising, considering that to interpret is to explain the meaning of something. Despite that, the related terms *interpretability* and *explainability* should not be used interchangeably, even though they are often mistaken for synonyms.

What is interpretability?

Interpretability is the extent to which humans, including non-subject-matter experts, can understand the cause and effect, and input and output, of a machine learning model. To say a model has a high level of interpretability means you can describe in a human-interpretable way its inference. In other words, why does an input to a model produce a specific output? What are the requirements and constraints of the input data? What are the confidence bounds of the predictions? Or, why does one variable have a more substantial effect than another? For interpretability, detailing how a model works is only relevant to the extent that it can explain its predictions and justify that it's the right model for the use case.

In this chapter's example, you could explain that there's a linear relationship between human height and weight, so using linear regression rather than a non-linear model makes sense. You can prove this statistically because the variables involved don't violate the assumptions of linear regression. Even when statistics are on our side, you still ought to consult with the domain knowledge area involved in the use case. In this one, we rest assured, biologically speaking, because our knowledge of human physiology doesn't contradict the connection between height and weight.

Beware of complexity

Many machine learning models are inherently harder to understand simply because of the math involved in the inner workings of the model or the specific model architecture. In addition to this, many choices are made that can increase complexity and make the models less interpretable, from dataset selection to feature selection and engineering, to model training and tuning choices. This complexity makes explaining how it works a challenge. Machine learning interpretability is a very active area of research, so there's still much debate on its precise definition. The debate includes whether total transparency is needed to qualify a machine learning model as sufficiently interpretable. This book favors the understanding that the definition of interpretability shouldn't necessarily exclude opaque models, which, for the most part, are complex, as long as the choices made don't compromise their trustworthiness. This compromise is what is generally called **post-hoc interpretability**. After all, much like a complex machine learning model, we can't explain exactly how a human brain makes a choice, yet we often trust its decision because we can ask a human for their reasoning. Post-hoc machine learning interpretation is exactly the same thing, except it's a human explaining the reasoning on behalf of the model. Using this particular concept of interpretability is advantageous because we can interpret opaque models and not sacrifice the accuracy of our predictions. We will discuss this in further detail in *Chapter 3*, *Interpretation Challenges*.

When does interpretability matter?

Decision-making systems don't always require interpretability. There are two cases that are offered as exceptions in research, outlined here:

- When incorrect results have no significant consequences. For instance, what if a machine learning model is trained to find and read the postal code in a package, occasionally misreads it, and sends it elsewhere? There's little chance of discriminatory bias, and the cost of misclassification is relatively low. It doesn't occur often enough to magnify the cost beyond acceptable thresholds.
- When there are consequences, but these have been studied sufficiently and validated enough in the real world to make decisions without human involvement. This is the case with a **traffic-alert and collision-avoidance system** (**TCAS**), which alerts the pilot of another aircraft that poses a threat of a mid-air collision.

On the other hand, interpretability is needed for these systems to have the following attributes:

- **Minable for scientific knowledge**: Meteorologists have much to learn from a climate model, but only if it's easy to interpret.
- **Reliable and safe**: The decisions made by a self-driving vehicle must be debuggable so that its developers can understand points of failure.
- **Ethical**: A translation model might use gender-biased word embeddings that result in discriminatory translations, but you must be able to find these instances easily to correct them. However, the system must be designed in such a way that you can be made aware of a problem before it is released to the public.
- **Conclusive and consistent**: Sometimes, machine learning models may have incomplete and mutually exclusive objectives—for instance, a cholesterol-control system may not consider how likely a patient is to adhere to the diet or drug regimen, or there might be a trade-off between one objective and another, such as safety and non-discrimination.

By explaining the decisions of a model, we can cover gaps in our understanding of the problem—*its incompleteness*. One of the most significant issues is that given the high accuracy of our machine learning solutions, we tend to increase our confidence level to a point where we think we fully understand the problem. Then, we are misled into thinking our solution covers *ALL OF IT*!

At the beginning of this book, we discussed how levering data to produce algorithmic rules is nothing new. However, we used to second-guess these rules, and now we don't. Therefore, a human used to be accountable, and now it's the algorithm. In this case, the algorithm is a machine learning model that is accountable for all of the ethical ramifications this entails. This switch has a lot to do with accuracy. The problem is that although a model may surpass human accuracy in aggregate, machine learning models have yet to interpret its results like a human would. Therefore, it doesn't second-guess its decisions, so as a solution it lacks a desirable level of completeness. and that's why we need to interpret models so that we can cover at least some of that gap. So, why is machine learning interpretation not already a standard part of the pipeline? In addition to our bias toward focusing on accuracy alone, one of the biggest impediments is the daunting concept of black-box models.

What are black-box models?

This is just another term for opaque models. A black box refers to a system in which only the input and outputs are observable, and you cannot see what is transforming the inputs into the outputs. In the case of machine learning, a black-box model can be opened, but its mechanisms are not easily understood.

What are white-box models?

These are the opposite of black-box models (see *Figure 1.3*). They are also known as transparent because they achieve total or near-total interpretation transparency. We call them **intrinsically interpretable** in this book, and we cover them in more detail in *Chapter 3*, *Interpretation Challenges*.

Have a look at a comparison between the models here:

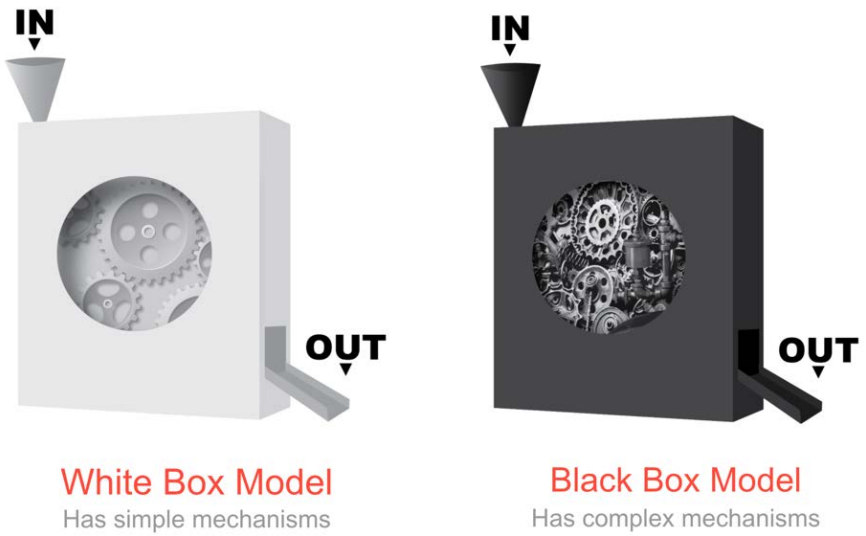

Figure 1.3 – Visual comparison between white- and black-box models

What is explainability?

Explainability encompasses everything interpretability is. The difference is that it goes deeper on the transparency requirement than interpretability because it demands human-friendly explanations for a model's inner workings and the model training process, and not just model inference. Depending on the application, this requirement might extend to various degrees of model, design, and algorithmic transparency. There are three types of transparency, outlined here:

- **Model transparency**: Being able to explain how a model is trained step by step. In the case of our simple weight prediction model, we can explain how the optimization method called **ordinary least squares** finds the β coefficient that minimizes errors in the model.
- **Design transparency**: Being able to explain choices made, such as model architecture and hyperparameters. For instance, we could justify these choices based on the size or nature of the training data. If we were performing a sales forecast and we knew that our sales had a seasonality of 12 months, this could be a sound parameter choice. If we had doubts, we could always use some well-established statistical method to find the right seasonality.
- **Algorithmic transparency**: Being able to explain automated optimizations such as grid search for hyperparameters; but note that the ones that can't be reproduced because of their random nature—such as random search for hyperparameter optimization, early stopping, and stochastic gradient descent—make the algorithm non-transparent.

Opaque models are called *opaque* simply because they lack *model transparency*, but for many models this is unavoidable, however justified the model choice might be. In many scenarios, even if you outputted the math involved in—say—training a neural network or a random forest, it would raise more doubts than generate trust. There are at least a few reasons for this, outlined here:

- **Not "statistically grounded"**: An opaque model training process maps an input to an optimal output, leaving behind what appears to be an arbitrary trail of parameters. These parameters are optimized to a cost function but are not grounded in statistical theory.
- **Uncertainty and non-reproducibility**: When you fit a transparent model with the same data, you always get the same results. On the other hand, opaque models are not equally reproducible because they use random numbers to initialize their weights or to regularize or optimize their hyperparameters, or make use of stochastic discrimination (such is the case for Random Forest).

- **Overfitting and the curse of dimensionality**: Many of these models operate in a high-dimensional space. This doesn't elicit trust because it's harder to generalize on a larger number of dimensions. After all, there's more opportunity to overfit a model, the more dimensions you add.
- **Human cognition and the curse of dimensionality**: Transparent models are often used for smaller datasets with fewer dimensions, and even if they aren't a transparent model, never use more dimensions than necessary. They also tend to not complicate the interactions between these dimensions more than necessary. This lack of unnecessary complexity makes it easier to visualize what the model is doing and its outcomes. Humans are not very good at understanding many dimensions, so using transparent models tends to make this much easier to understand.
- **Occam's razor**: This is what is called the principle of simplicity or parsimony. It states that the simplest solution is usually the right one. Whether true or not, humans also have a bias for simplicity, and transparent models are known for— if anything—their simplicity.

Why and when does explainability matter?

Trustworthy and ethical decision-making is the main motivation for interpretability. Explainability has additional motivations such as causality, transferability, and informativeness. Therefore, there are many use cases in which total or nearly total transparency is valued, and rightly so. Some of these are outlined here:

- **Scientific research**: Reproducibility is essential to the scientific method. Also, using statistically grounded optimization methods is especially desirable when causality needs to be proven.
- **Clinical trials**: These must also produce reproducible findings and be statistically grounded. In addition to this, given the potential gravity of overfitting, they must use the fewest dimensions possible and models that don't complicate them.
- **Consumer product safety testing**: Much as with clinical trials, when life-and-death safety is a concern, simplicity is preferred whenever possible.

- **Public policy and law**: This is a more nuanced discussion, as part of what is called by law scholars **algorithmic governance**, and they have distinguished between **fishbowl transparency** and **reasoned transparency**. The former is closer to the rigor required for consumer product safety testing, and the latter is one where post-hoc interpretability would suffice. One day, the government could be entirely run by algorithms. When that happens, it's hard to tell which policies will align with which form of transparency, but there are many areas of public policy, such as criminal justice, where absolute transparency is necessary. However, whenever total transparency contradicts privacy or security objectives, a less rigorous form of transparency would have to make do.
- **Criminal investigation and regulatory compliance audits**: If something goes wrong, such as an accident at a chemical factory caused by a robot malfunction or a crash by an autonomous vehicle, an investigator needs to trace the **decision trail**. This is to "facilitate the assignment of accountability and legal liability". Even when no accident has happened, this kind of auditing can be performed when mandated by authorities. Compliance auditing applies to industries that are regulated, such as financial services, utilities, transportation, and healthcare. In many cases, fishbowl transparency is preferred.

A business case for interpretability

This section describes several practical business benefits for machine learning interpretability, such as better decisions, as well as being more trusted, ethical, and profitable.

Better decisions

Typically, machine learning models are trained and then evaluated against the desired metrics. If they pass quality control against a hold-out dataset, they are deployed. However, once tested in the real world, that's when things can get wild, as in the following hypothetical scenarios:

- A high-frequency trading algorithm could single-handedly crash the stock market.
- Hundreds of smart home devices might inexplicably burst into unprompted laughter, terrifying their users.
- License-plate recognition systems could incorrectly read a new kind of license plate and fine the wrong drivers.

- A racially biased surveillance system could incorrectly detect an intruder, and because of this guards shoot an innocent office worker.
- A self-driving car could mistake snow for a pavement, crash into a cliff, and injure passengers.

Any system is prone to error, so this is not to say that interpretability is a cure-all. However, focusing on just optimizing metrics can be a recipe for disaster. In the lab, the model might generalize well, but if you don't know why the model is making the decisions, then you can miss on an opportunity for improvement. For instance, knowing *what* the self-driving car thinks is a road is not enough, but knowing *why* could help improve the model. If, say, one of the reasons was that road is light-colored like the snow, this could be dangerous. Checking the model's assumptions and conclusions can lead to an improvement in the model by introducing winter road images into the dataset or feeding real-time weather data into the model. Also, if this doesn't work, maybe an algorithmic fail-safe can stop it from acting on a decision that it's not entirely confident about.

One of the main reasons why a focus on machine learning interpretability leads to better decision-making was mentioned earlier when we talked about completeness. If you think a model is complete, what is the point of making it better? Furthermore, if you don't question the model's reasoning, then your understanding of the problem must be complete. If this is the case, perhaps you shouldn't be using machine learning to solve the problem in the first place! Machine learning creates an algorithm that would otherwise be too complicated to program in *if-else* statements, precisely to be used for cases where our understanding of the problem is incomplete!

It turns out that when we predict or estimate something, especially with a high level of accuracy, we think we control it. This is what is called the **illusion of control bias**. We can't underestimate the complexity of a problem just because, in aggregate, the model gets it right almost all the time. Even for a human, the difference between snow and concrete pavement can be blurry and difficult to explain. How would you even begin to describe this difference in such a way that it is always accurate? A model can learn these differences, but it doesn't make it any less complex. Examining a model for points of failure and continuously being vigilant for outliers requires a different outlook, whereby we admit that we can't control the model but we can try to understand it through interpretation.

The following are some additional decision biases that can adversely impact a model, and serve as reasons why interpretability can lead to better decision-making:

- **Conservatism bias**: When we get new information, we don't change our prior beliefs. With this bias, entrenched pre-existing information trumps new information, but models ought to evolve. Hence, an attitude that values questioning prior assumptions is a healthy one to have.
- **Salience bias**: Some prominent or more visible things may stand out more than others, but statistically speaking, they should get equal attention to others. This bias could inform our choice of features, so an interpretability mindset can expand our understanding of a problem to include other less perceived features.
- **Fundamental attribution error**: This bias causes us to attribute outcomes to behavior rather than circumstances, character rather than situations, nature rather than nurture. Interpretability asks us to explore deeper and look for the less obvious relationships between our variables or those that could be missing.

One crucial benefit of model interpretation is locating *outliers*. These outliers could be a potential new source of revenue or a liability waiting to happen. Knowing this can help us to prepare and strategize accordingly.

More trusted brands

Trust is defined as a belief in the reliability, ability, or credibility of something or someone. In the context of organizations, trust is their reputation; and in the unforgiving court of public opinion, all it takes is one accident, controversy, or fiasco to lose substantial amounts of public confidence. This, in turn, can cause investor confidence to wane.

Let's consider what happened to Boeing after the 737 MAX debacle or Facebook after the 2016 presidential election scandal. In both cases, there were short-sighted decisions solely made to optimize a single metric, be it forecasted plane sales or digital ad sales. These underestimated known potential points of failure and missed out entirely on very big ones. From there, it can often get worse when organizations resort to fallacies to justify their reasoning, confuse the public, or distract the media narrative. This behavior might result in additional public relations blunders. Not only do they lose credibility with *what they do* with their first mistake but they attempt to fool people, losing credibility with *what they say*.

And these were examples of, for the most part, decisions made by people. With decisions made exclusively by machine learning models, this could get worse because it is easy to drop the ball and keep the accountability in the model's corner. For instance, if you started to see offensive material in your Facebook feed, Facebook could say it's because its model was trained with *your data* such as your comments and likes, so it's really a reflection of *what you want to see*. Not their fault—your fault. If the police targeted your neighborhood for aggressive policing because it uses PredPol, an algorithm that predicts where and when crimes will occur, it could blame the algorithm. On the other hand, the makers of this algorithm could blame the police because the software is trained on their police reports. This generates a potentially troubling feedback loop, not to mention an accountability gap. And if some pranksters or hackers eliminate lane markings, this could cause a Tesla self-driving car to veer into the wrong lane. Is this Tesla's fault that they didn't anticipate this possibility, or the hackers', for throwing a monkey wrench into their model? This is what is called an **adversarial attack**, and we discuss this in *Chapter 13, Adversarial Robustness*.

It is undoubtedly one of the goals of machine learning interpretability to make models better at making decisions. But even when they fail, you can show that you tried. Trust is not lost entirely because of the failure itself but because of the lack of accountability, and even in cases where it is not fair to accept all the blame, some accountability is better than none. For instance, in the previous set of examples, Facebook could look for clues as to why offensive material is shown more often, then commit to finding ways to make it happen less even if this means making less money. PredPol could find other sources of crime-rate datasets that are potentially less biased, even if they are smaller. They could also use techniques to mitigate bias in existing datasets (these are covered in *Chapter 11, Bias Mitigation and Causal Inference Methods*). And Tesla could audit its systems for adversarial attacks, even if this delays shipment of its cars. All of these are interpretability solutions. Once a common practice, they can lead to an increase in not only public trust—be it from users and customers, but also internal stakeholders such as employees and investors.

The following screenshot shows some public relation AI blunders that have occurred over the past couple of years:

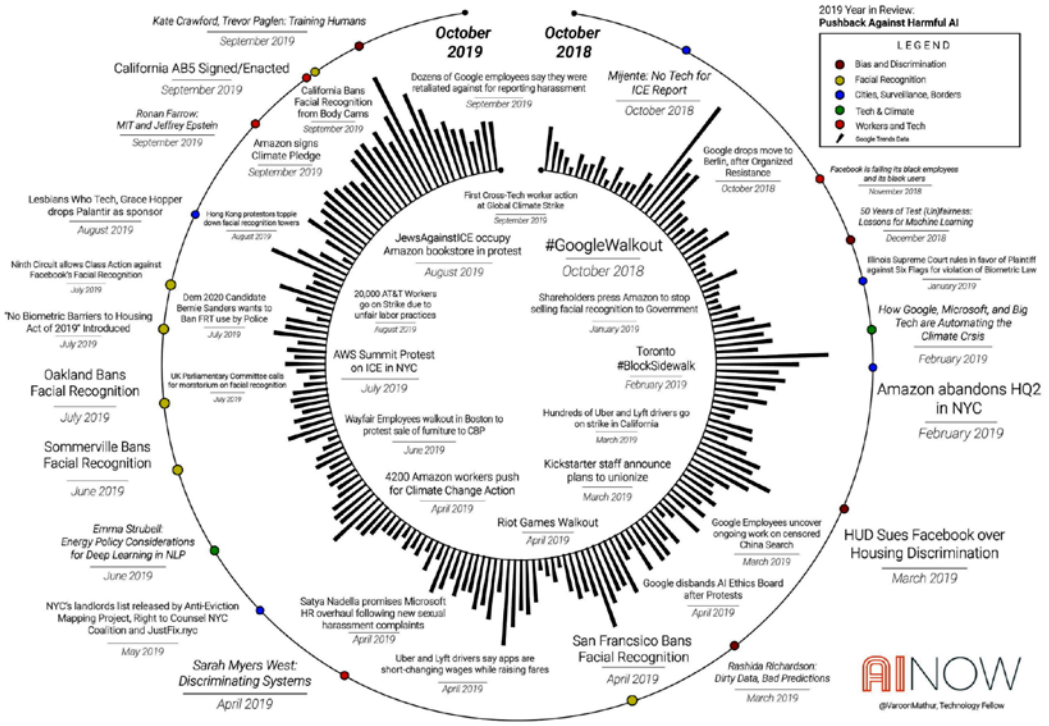

Figure 1.4 – AI Now Institute's infographic with AI's public relation blunders for 2019

Due to trust issues, many AI-driven technologies are losing public support, to the detriment of both companies that monetize AI and users that could benefit from them (see *Figure 1.4*). This, in part, requires a legal framework at a national or global level and, at the organizational end, for those that deploy these technologies, more accountability.

More ethical

There are three schools of thought for ethics: utilitarians focus on consequences, deontologists are concerned with duty, and teleologicalists are more interested in overall moral character. So, this means that there are different ways to examine ethical problems. For instance, they are useful lessons to draw from all of them. There are cases in which you want to produce the greatest amount of "good", despite some harm being produced in the process. Other times, ethical boundaries must be treated as lines in the sand you mustn't cross. And at other times, it's about developing a righteous disposition, much like many religions aspire to do. Regardless of the school of ethics we align with, our notion of what it is evolves with time because it mirrors our current values. At this moment, in Western cultures, these values include the following:

- Human welfare
- Ownership and property
- Privacy
- Freedom from bias
- Universal usability
- Trust
- Autonomy
- Informed consent
- Accountability
- Courtesy
- Environmental sustainability

Ethical transgressions are cases whereby you cross the moral boundaries that these values seek to uphold, be it by discriminating against someone or polluting their environment, whether it's against the law or not. Ethical dilemmas occur when you have a choice between options that lead to transgressions, so you have to choose between one and another.

The first reason machine learning is related to ethics is because technologies and ethical dilemmas have an intrinsically linked history.

Since the first widely adopted tool made by humans, it brought progress but also caused harm, such as accidents, war, and job losses. This is not to say that technology is always bad but that we lack the foresight to measure and control its consequences over time. In AI's case, it is not clear what the harmful long-term effects are. What we can anticipate is that there will be a major loss of jobs and an immense demand for energy to power our data centers, which could put stress on the environment. There's speculation that AI could create an "algocratic" surveillance state run by algorithms, infringing on values such as privacy, autonomy, and ownership.

The second reason is even more consequential than the first. It's that machine learning is a technological first for humanity: machine learning is a technology that can make decisions for us, and these decisions can produce individual ethical transgressions that are hard to trace. The problem with this is that accountability is essential to morality because you have to know who to blame for human dignity, atonement, closure, or criminal prosecution. However, many technologies have accountability issues to begin with, because moral responsibility is often shared in any case. For instance, maybe the reason for a car crash was partly due to the driver and mechanic and car manufacturer. The same can happen with a machine learning model, except it gets trickier. After all, a model's programming has no programmer because the "programming" was learned from data, and there are things a model can learn from data that can result in ethical transgressions. Top among them are biases such as the following:

- **Sample bias**: When your data, the sample, doesn't represent the environment accurately, also known as the population
- **Exclusion bias**: When you omit features or groups that could otherwise explain a critical phenomenon with the data
- **Prejudice bias**: When stereotypes influence your data, either directly or indirectly
- **Measurement bias**: When faulty measurements distort your data

Interpretability comes in handy to mitigate bias, as seen in *Chapter 11, Bias Mitigation and Causal Inference Methods*, or even place guardrails on the right features, which may be a source of bias. This is covered in *Chapter 12, Monotonic Constraints and Model Tuning for Interpretability*. As explained in this chapter, explanations go a long way in establishing accountability, which is a moral imperative. Also, by explaining the reasoning behind models, you can find ethical issues before they cause any harm. But there are even more ways in which models' potentially worrisome ethical ramifications can be controlled for, and this has less to do with interpretability and more to do with design. There are frameworks such as **human-centered design**, **value-sensitive design**, and **techno moral virtue ethics** that can be used to incorporate ethical considerations into every technological design choice. An article by Kirsten Martin (`https://doi.org/10.1007/s10551-018-3921-3`) also proposes a specific framework for algorithms. This book won't delve into algorithm design aspects too much, but for those readers interested in the larger umbrella of ethical AI, this article is an excellent place to start. You can see Martin's algorithm morality model in *Figure 1.5* here:

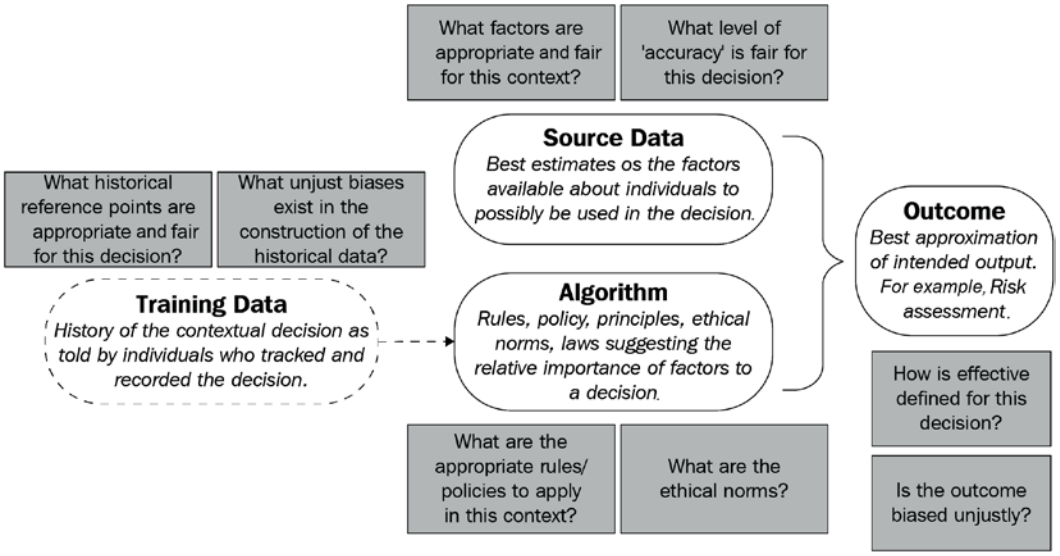

Figure 1.5 – Martin's algorithm morality model

Organizations should take the ethics of algorithmic decision-making seriously because ethical transgressions have monetary and reputation costs. But also, AI left to its own devices could undermine the very values that sustain democracy and the economy that allows businesses to thrive.

More profitable

As seen already in this section, interpretability improves algorithmic decisions, boosting trust and mitigating ethical transgressions.

When you leverage previously unknown opportunities and mitigate threats such as accidental failures through better decision-making, you can only improve the bottom line; and if you increase trust in an AI-powered technology, you can only increase its use and enhance overall brand reputation, which also has a beneficial impact on profits. On the other hand, as for ethical transgressions, they can be there by design or by accident, but when they are discovered, they adversely impact both profits and reputation.

When businesses incorporate interpretability into their machine learning workflows, it's a virtuous cycle, and it results in higher profitability. In the case of a non-profit or governments, profits might not be a motive. Still, finances are undoubtedly involved because lawsuits, lousy decision-making, and tarnished reputations are expensive. Ultimately, technological progress is contingent not only on the engineering and scientific skills and materials that make it possible but its voluntary adoption by the general public.

Summary

Upon reading this chapter, you should now have a clear understanding of what machine learning interpretation is and isn't, and recognize the importance of interpretability. In the next chapter, we will learn what can make machine learning models so challenging to interpret, and how you would classify interpretation methods in both category and scope.

Image sources

- Mathur, Varoon (2019). *AI in 2019: A Year in Review - The Growing Pushback Against Harmful AI*. AI Now Institute via Medium.
- Martin, K. (2019). *Ethical Implications and Accountability of Algorithms*. Journal of Business Ethics 160. 835–850. `https://doi.org/10.1007/s10551-018-3921-3`

Further reading

- Microsoft (2019). *Responsible AI principles from Microsoft*. Retrieved from `https://www.microsoft.com/en-us/ai/responsible-ai`
- Lipton, Zachary (2017). *The Mythos of Model Interpretability*. _ICML 2016 Human Interpretability in Machine Learning Workshop_ `https://doi.org/10.1145/3236386.3241340`

- Doshi-Velez, F. & Kim, B. (2017). *Towards A Rigorous Science of Interpretable Machine Learning*. http://arxiv.org/abs/1702.08608
- Roscher, R., Bohn, B., Duarte, M.F. & Garcke, J. (2020). *Explainable Machine Learning for Scientific Insights and Discoveries*. IEEE Access, 8, 42200-42216. https://dx.doi.org/10.1109/ACCESS.2020.2976199
- Coglianese, C. & Lehr, D. (2019). *Transparency and algorithmic governance*. Administrative Law Review, 71, 1-4. https://ssrn.com/abstract=3293008
- Weller, Adrian. (2019) "*Transparency: Motivations and Challenges*". arXiv:1708.01870 [Cs]. http://arxiv.org/abs/1708.01870