CHAPTER

6

Creating the Model

Where the AI magic happens

The Rubik's Cube, which is the 3-D puzzle that stirred up a craze during the 1980s, continues to challenge many people. But data scientists at OpenAI saw it as an interesting game to learn more about AI. In October 2019, the company announced it created a model that used a robotic hand to solve the Rubik's Cube. You can find the video on YouTube (`https://bit.ly/3f5fuy4`).

What's interesting about this is that the robotic hand was not cutting-edge. It had been developed 15 years earlier. Rather, OpenAI wanted to show how powerful algorithms can transform physical systems.

The company's researchers used deep learning, reinforcement learning, and Kociemba's algorithm. They used simulations for training, which got more and more complicated. For the most part, this is how the AI system was able to learn to solve the Rubik's Cube. This involved not only understanding the enormous number of combinations— 43,252,003,274,489,856,000 combinations, or 43 quintillion—but also the use of computer vision and coordination with the robotic hand. According to the OpenAI blog: "We focus on the problems that are currently difficult for machines to master: perception and dexterous manipulation."[1]

[1] `https://openai.com/blog/solving-rubiks-cube/`

T. Taulli, *Implementing AI Systems*, https://doi.org/10.1007/978-1-4842-6385-3_6

The end result: The AI was able to solve the Rubik's Cube about 60% of the time, depending on the environment. No doubt, OpenAI will continue to push the innovation to improve the accuracy rate.

This example illustrates how amazing AI can be. But it also shows the inherent power of algorithms. They can definitely be transformative.

True, for most organizations, there isn't the luxury of this kind of experimentation. But AI models do hold the potential for making notable improvements and can certainly be a key driver for business performance.

So in this chapter, we'll take a look at how to create solid AI models.

Model Selection

When it comes to model selection, it's a good idea to be creative and experiment with a variety of algorithms (keep in mind that there are hundreds available). This process has become much easier because of the availability of open source and proprietary AI platforms (we will cover some of them later in this chapter).

Yet there is a temptation to just focus on the accuracy. But this can lead to outcomes that are far from ideal. How so? For example, suppose that Model A has an accuracy rate of 85% and Model B is at 80%. However, Model A takes 10 times more time to train and is much more expensive. So is the extra 5% really that important? For many business applications, the answer may actually be "not very."

There are other issues to take under consideration when evaluating a model:

- How much data does it need? Does your organization have the right kind? Is there enough quality data?
- Can the model be explained? This is very important. When rolling out AI in a company, there can be much resistance. But if the model makes inherent sense to those who are non-technical, then adoption may be easier.
- How hard is it to maintain and deploy the model?
- Is there a need for a third-party audit?

It's a good idea to first test out simple models. You may realize that there is no need for something like a neural network or deep learning. Instead, the results from, say a regression model, may be good enough for your project.

Also, by starting with simpler models, you can establish baselines. This will help provide structure for the evaluation process.

Now even though software systems can help with the model selection process, the expertise of an experienced data scientist can go a long way. Such a person can often find a set of models that meet your requirements.

"As soon as the problem becomes more complicated because of a more complex task, messier data, or more innovative required solutions, then classic prepackaged or automated applications might not be enough any longer," said Rosaria Silipo, who is a PhD and a Principal Data Scientist at KNIME. "Some more creative and innovative thinking may be necessary, and your experience with past projects could be the key to the final solution. So data science is still kind of an art, where experience and knowledge play a determining role to find the optimal, and sometimes the only, possible solution."[2]

But of course, when selecting a model, there needs to be a focus on the problem to be solved. Here are some examples:

- If you are looking at categorical questions, such as "Is this a risky customer?" then you might want to look at algorithms for classification and perhaps clustering.
- If you are making a prediction for a numeric value, say for the value of a home, then a regression model might be a good option.
- If you have a set of time-series data for the customer journey, then you could employ a long short term memory (LSTM) network or a recurrent neural network (RNN).
- If you are working on a computer vision application, then you would look at RNNs and convolutional neural networks (CNNs).
- If you are developing a natural language processing (NLP) application, then you would consider bag of words, the Naive Bayes algorithm, topic modeling, and LSTM.

So what about creating unique AI models? This is definitely something that can be incredibly powerful. Some companies will even seek patents on their models. But it's important to note that you will likely need PhDs on your staff who have a deep understanding of AI. This is not practicable for most companies. But the good news is that publicly available models should be more than sufficient for most projects.

[2] From the author's interview with Rosaria Silipo on June 12, 2020.

Despite the different approaches, it's important to note that there is no right or wrong model. Rather, it's about finding a model that is appropriate for your business goals and the underlying data. Actually, it's common that several models work just fine.

But consider that in model building there is a well-known concept called the No Free Lunch Theorem, which means that no model is best for all tasks. There will always be some inherent limitations.

Ensemble Models

With the model selection process, there may be occasions when there should be more than one used. This is known as an ensemble model.

This does add to the complexity. But if an ensemble model is done properly, the results can be robust.

An example of this is Netflix. Back in 2006, the company established the Netflix Prize, which offered $1 million for any person or team that could improve the accuracy of the recommendation engine by at least 10%. To this end, the company open sourced a data set that had more than 100 million ratings of 17,770 movies from 480,189 users. It was a gold mine for ambitious data scientists!

At the time, Netflix was actually having challenges with its own models. It was essentially reaching diminishing returns. So why not try crowdsourcing?

The contest did inspire many people to participate, and there were lots creative solutions. Yet the goal proved to be a challenge.

However, a few years later, a team called BellKor's Pragmatic Chaos won the contest. To do this, they created a baseline model that helped to mitigate some of the problems with the data. After all, some movies had sparse ratings while others had large numbers. There were also some users who would always give low ratings and others who would just provide top ones! Like any data set, it was messy and required considerable normalization.

But there was another hurdle: the model testing. All in all, the team faced some tough issues. For example, an algorithm may continue to recommend the same films, there may not be the right genre for a particular firm, or there could be changes in the ratings over time (as societal tastes or attitudes evolve).

To deal with all this, the team used ensemble modeling. In fact, this meant using hundreds of algorithms. But in the end, the approach worked extremely well.

Training the Model

After you have identified the different algorithms for the project, you will then want to train them. Basically, can they learn and provide sufficient predictions?

You do this by using a data set. The first step is to randomize the information. By doing this, there will be less likelihood that the model will detect false patterns. Next, you need to divide the data sets into different sections, which will help to provide more accurate outcomes.

So let's take a look at the process.

Phase 1: Testing

This will involve anywhere from 70% to 80% of the data set. As you apply this data to the different models, there will often be varying results. Some will be fairly useless. But when it comes to modeling, there needs to be considerable trial and error.

In this phase, you will have to engage in some fine tuning of the model, such as with the parameters (each new iteration is called a "training step"). We'll look at the process later in this chapter.

Phase 2: Validation

This is where you have a sample of 10% to 20% of the data set. Given that the models have been tweaked, there should be an improvement in the accuracy. But in this stage, there should be analysis to see if there are problems like

- *Overfitting*: This is where the model is essentially memorizing patterns and has not effectively learned from the data. One of the signs of this is if there is a very high accuracy rate, say over 90%. So how can the overfitting be reduced? One approach is to collect more diverse data, which should help provide for a more robust model. What's more, using a less complicated algorithm could help with the overfitting.
- *Gaps*: The validation may show that the model has difficulties recognizing certain elements. For example, if you have an NLP app, one potential problem is that the AI will not be able to detect various words. If this is the case, then you probably need to add new features.
- *Underfitting*: This is where the model does not adequately reflect reality. Some ways to deal with this include increasing the number of parameters or use a more advanced algorithm.

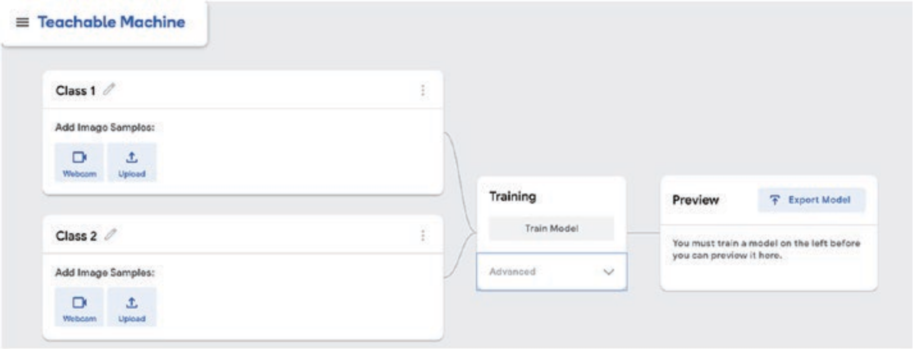
Phase 3: Holdout Set

This is also called the “testing data.” It includes 5% to 10% of the data set. Regardless of the name, this phase will give a final assessment of the overall accuracy of the model. The goal is to get a sense of how the AI reacts in a real-world environment.

Cloud-Based Model Systems

The training of an AI model does involve complex coding, such as with Python. While we will not show how this is done in this book, since the focus is for readers who are non-technical, there are still simple ways to build models.

Take a look at Teachable Machine from Google (https://teachablemachine.withgoogle.com/). By using drag-and-drop and pull-down menus, you can create your own AI model. First, you gather the data, such as from your photos. You can even upload audio files. Then you train the model. Note that Teachable Machine provides a myriad of options to tune the parameters to get better results. Finally, you can export the model and even host it on the Internet for free. Figure 6-1 shows what the app looks like.



You must train a model on the left before you can preview it here.

Figure 6-1. This is the interactive model creation system called Teachable Machine

Another interesting AI tool is the Machine Learning Playground (https://ml-playground.com/). Similar to the Teachable Machine, the workflow is easy. But the Machine Learning Playground provides a myriad of different models to apply to your data sets like k-nearest neighbors, support vector machines, artificial neural networks, and so on. Figure 6-2 shows what this looks like.

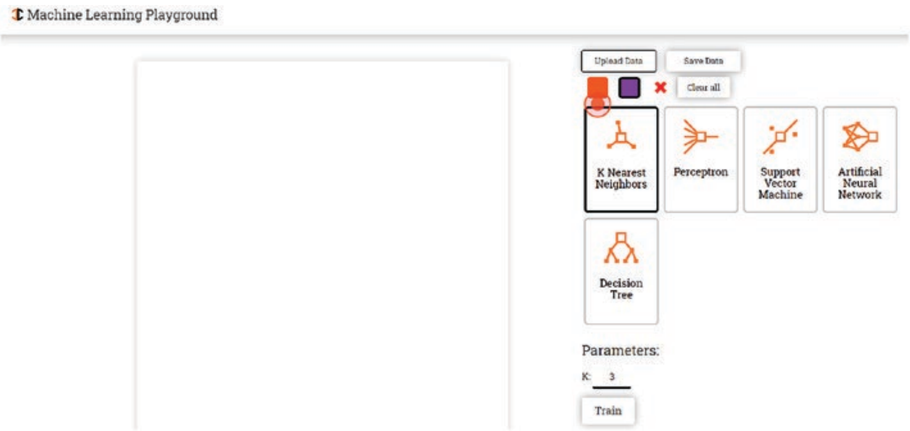


Figure 6-2. The Machine Learning Playground is an AI model creation system

Feature Engineering

After you have identified one or more models for the AI project, it's time for feature engineering. Keep in mind that this is the most important part of the model development process. Even a few wrong moves can mean that the results will be subpar.

To make any model work, you need to find those values that help predict results or learn new patterns. These values are the features.

Let's take an example. Suppose you want to predict the weather and you have a large data set for this. What are the features? Some data to consider would be the temperature, barometric readings, and perhaps radar information.

As you can see, this process involves some understanding of the domain. Thus, when it comes to feature engineering, an approach is to have experts help out (in this example, this would probably be a meteorologist). They should be able to quickly come up with the kinds of metrics and variables that are material. This process is called hand-crafting features.

But this does not mean feature engineering should only be for experts. Having a brainstorming session with the team can also be a good idea. Hey, even experts can be too narrow and miss important patterns.

In certain cases, the data will be non-mathematical. This means you will need to do a conversion. An example is if you are creating a model to determine if a tumor is malignant or benign. You may identify features such as tumor size and tumor shape.

Seems straightforward? Not necessarily. Features can easily be fuzzy. In this example, the tumor size could be expressed in terms of radius, surface area, or weight. Oh, and for shape, there are even more ways to look at this. You ultimately have to decide on the types of features and then have a number that represents them.

As this example shows, the complexity can increase greatly when using feature engineering. The process also is generally time-consuming.

Another issue is that the features usually are not transferable. What does this mean? With this example, the measurements for lung tumors will not be the same for breast cancer. This can add to the challenges of creating models.

Consider that sophisticated deep learning models can actually be used for feature engineering. The algorithms will crunch the numbers across many hidden layers to come up with patterns, often which humans are unable to detect. However, using deep learning may still not be enough. Having a human in the loop is usually advisable for feature engineering.

Deep learning models can also lead to terrible results. Sometimes they may be downright comical! Take the following real-world example from Sheldon Fernandez, who is the CEO of DarwinAI: "One of our automotive clients encountered some bizarre behavior in which a self-driving car would turn left with increasing regularity when the sky was a certain shade of purple. After months of painful debugging, they determined the training for certain turning scenarios had been conducted in the Nevada desert when the sky was a particular hue. Unbeknownst to its human designers, the neural network had established a correlation between its turning behavior and the celestial tint."[3]

Note In feature engineering, you often will have two variables that are highly correlated. The best practice is to either drop one or combine the two so as to avoid inaccurate outcomes.

Parameters vs. Hyperparameters

The words "parameters" and "hyperparameters" are often used interchangeably. But this is a mistake. They each have important distinctions with AI models. But as should be no surprise, the terms can be very confusing.

A parameter is a variable that is internal to the model and whose value is determined from the underlying data. Then what are some examples? One would be the weights for a deep learning model. These values are optimized using algorithms.

[3] From the author's interview with Sheldon Fernandez on December 11, 2018.

As for a hyperparameter, this is a value that is external to the model and cannot be estimated from the data. How so? The reason is that a data scientist will specify the hyperparameters (the process is called hyperparameter tuning), such as by using a heuristic (that is, a "rule of thumb"). For example, the k-nearest neighbor classification algorithm requires the use of hyperparameters because there is no approach to calculate the exact value.

There are many machine learning programs that can provide automated hyperparameter tuning. This involves indicating the variables you want to have variation on and setting a metric as what thresholds to achieve. The system will then optimize the hyperparameters, which often use search algorithms like Bayesian optimization, grid search, and random search.

Accuracy

In the summer of 2018, the ACLU published a blockbuster blog post. It showed that Amazon.com's facial recognition software, called Rekognition, had some glaring deficiencies. The ACLU ran the system against photos of the members of the U.S. Congress against a database of mugshots. The outcome? 28 were identified as having been arrested for crimes! The false results disproportionately favored minorities, including Congressman John Lewis. For the ACLU, it only had to shell out $12.33 to use the model.

According to the blog post: "An identification—whether accurate or not—could cost people their freedom or even their lives. People of color are already disproportionately harmed by police practices, and it's easy to see how Rekognition could exacerbate that. A recent incident in San Francisco provides a disturbing illustration of that risk. Police stopped a car, handcuffed an elderly Black woman, and forced her to kneel at gunpoint—all because an automatic license plate reader improperly identified her car as a stolen vehicle."[4]

It's really chilling—and shows how AI accuracy can be high stakes. It is also an important reminder that the technology still has a long way to go.

So when evaluating accuracy, there needs to be much thought on the consequences. True, there will be many cases where you do not need high levels, such as when calculating churn or conversation rates for sales.

On the other hand, there are some areas where accuracy is perhaps the most important metric. This is usually the case for areas like medicine, finance, autonomous vehicles, and so on. If you are diagnosing for cancer, the AI system better be right. Likewise, if a person goes to an ATM, an accuracy rate of 99% would be completely unacceptable.

[4] www.aclu.org/blog/privacy-technology/surveillance-technologies/amazons-face-recognition-falsely-matched-28

Earlier in this chapter, you looked at how to divide the data into different parts in order to train the model. Keep in mind that, in this process, there will be an evaluation of the accuracy rates. This will also be the case when engaging in feature engineering.

Algorithms will have their own usual accuracy measures. You saw this in Chapter 2. For example, with a regression analysis, you could use the standard error and R-squared. Or if you have a logistic regression, you can put together a confusion matrix.

For accuracy, you will also see the following terms: precision, recall and F1 score. Let's take a look at each of them. First, let's set up a matrix, as seen in Figure 6-1.

Table 6-1. A Matrix to Illustrate Precision, Recall, and the F1 Score

	Predicted		
Actual		Yes	No
	Yes	True positive	False negative
	No	False positive	True negative

In the above, a true positive and true negative are when an outcome is a correct prediction. And yes, the false positives and false negatives are those with the incorrect predictions.

These concepts can definitely get fuzzy. So then, let's get a fuller explanation. Suppose we have a model that will output Yes if a tumor is cancerous and No if the tumor is benign. Then we process an X-ray and it outputs Yes, and when we check the patient's vitals, the person does have cancer. This would be a true positive since the model predicted Yes and the actual value was Yes.

But suppose we try another X-ray and we get something different. That is, the model predicts that there is no cancer and the actual value is No.

Well, when it comes to this analysis, we want true positives and true negatives because they are accurate predictions. On the other hand, we want to avoid these:

- *False positives*: This is where the model indicates cancer but the patient actually does not have the disease.
- *False negatives*: With this, the model predicts no cancer but the person does have the disease.

Given our matrix then, we can compute various metrics. One is the simplest: accuracy. This shows the number of correct predictions divided by the total predictions or

(True positives + true negatives)/(True positives + true negatives + false negatives + true negatives)

However, the accuracy metric can be misleading if there is a disproportionate number of false positives or false negatives. This is why we have metrics like the following:

Precision

This is calculated as follows:

True positives/(True positives + false positives)

This essentially takes into the account the false positives. Thus, if the accuracy is fairly high, say over 60%, then there is a lower risk of false positives.

Recall

This is computed as the following:

True positive/(True positive + false negative)

Yes, this accounts for the false negatives. So if the accuracy is high, then there is less of a likelihood of false negatives.

F1 Score

This is the weighted average of the precision and recall computations. In other words, the F1 score metric is more comprehensive, giving an overall indication of the false positives and false negatives.

The use of precision, recall and the F1 score can be critical in certain use cases. This is especially the situation with medical analysis. For instance, if you are diagnosing a potentially life-threatening disease, then you may want to allow for having false negatives but no false positives. This is why this type of analysis will involve having an expert review for all the results. By combining AI with a physician's expert judgement, there are often better results.

While all these approaches are quite useful, sometimes it is important to get a sample of some of the incorrect predictions. Was the result downright egregious? Did the model mistake a lightbulb for a TV? If so, there could be serious flaws with the algorithms even though the accuracy rate may be high.

AI Tools

According to G2.com, there are nearly 100 platforms to help build AI models. So it is really impossible to check them all out. And there are also new ones popping up.[5]

Many of the tools are also open source. This means that the software is free to use, although some of the systems may have restrictions and premium versions.

Yet this does not mean you should avoid proprietary platforms. Consider that these solutions may be more intuitive and automated. They may also have stronger backing, such as from venture capitalists.

So in the next part of this chapter, we will look at both open source and proprietary platforms.

Open Source Tools

With open source AI platforms, you will usually work with a computer language. By far, the most popular is Python. It is the fastest growing programing language in the world.

The mastermind of Python is Guido van Rossum, who launched the system in late 1989. The timing was spot on. The Internet was starting to emerge, especially in the academic community. The language was also fairly easy to learn and worked seamlessly with statistics and machine learning.

■ **Note** van Rossum named Python after his favorite comedy, ***Monty Python's Flying Circus***.

But there are other reasons for the huge success of Python. One is that it became a must-have for the global academic community. And there was strong adoption from businesses and startups.

The open source model has also been essential. This has meant that thousands of developers have contributed to the evolution of the language, such as with new features, packages, and add-ons.

[5] www.g2.com/categories/data-science-and-machine-learning-platforms

Some of the popular add-ons, especially for machine learning and AI, include

- *NumPy*: This is an advanced system that helps to develop extensive indexes, matrices, slicing, tensors, masking, and multi-dimensional arrays. NumPy is written in both Python and C, which has helped improve the speed and efficiency. Note that the system has been optimized to be used with GPUs.
- *Pandas*: This add-on is focused on data analysis and cleanup as well as visualizations. This system also uses the C language for better performance.

A common way to work with Python is to use a Jupyter Notebook. This is a web app that allows for coding in the language in a structured manner (the commands are entered in cells). The code and scripts can then be shared with other programmers or made publicly available.

Besides Python, there is the R language. Its origins go back to 1993, when Ross Ihaka and Robert Gentleman teamed up to create a platform that would work better with statistics and visualizations. R is widely used for AI and some of the customers include Google, Airbnb, Uber, and Facebook.

OK then, now let's take a look at some of the top open source AI platforms:

TensorFlow

As deep learning became an important part of AI, Google realized it needed to use this technology. Its massive data sets were ideal for this and there was a need to provide more automation for its huge platforms.

But there was a nagging issue: There were no effective tools to create the deep learning models.

So Google developed its own, which was called TensorFlow. The project started in 2011 as a part of the Google Brain division. TensorFlow worked quite well and proved critical for next-generation apps.

But Google did something else that was noteworthy. In 2015, the company made TensorFlow open source. Why give it away? A major reason for this was that Google wanted to accelerate the innovation of AI. And this certainly happened—and quickly. The result is that TensorFlow has turned into an industry standard.

Consider that you can program TensorFlow in R, Swift, and JavaScript. But Python is the most popular.

TensorFlow operates by creating a tensor structure, which is essentially a multidimensional array, and is able to process the flow of the data on a graph. Since it powers applications for Google (like the Photos app), TensorFlow is robust for the most intense environments, whether on the desktop, mobile, edge devices, or clusters.

PyTorch

Facebook is the developer of PyTorch, which was launched in early 2016. The initial focus was on applications for computer vision and NLP. Given the rapid evolution of the platform, PyTorch has become a worthy rival of TensorFlow.

PyTorch was built to allow quick iterations, which is important for the model creation process. This is why the platform has become quite popular for researchers. But the system is also user-friendly and leverages the resources of Python.

Keras

Keras is an AI platform that is geared primarily to beginners. With just a few lines of code, you can create a sophisticated neural network. Even experts can use Keras as it can be good for quick experiments and prototypes.

According the website: "Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent and simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable error messages. It also has extensive documentation and developer guides."[6]

Despite the simplicity, Keras is still powerful. Consider that TensorFlow has integrated the technology in its own platform.

Scikit-Learn

This is a full-blown AI platform, providing for the development of supervised and unsupervised models. Like Keras, the interface is easy to use. Yet the technology is strong enough for deploying enterprise applications.

Scikit-Learn is one of the older AI platforms; the project was started in 2007. It was part of an initiative from David Cournapeau for the Google Summer Code program. By the following year, Matthieu Brucher took on Scikit-Learn for his thesis. Then in 2010 the platform was publicly released. Since then, there has been a release cycle of about every three months.[7]

[6] https://keras.io/

[7] https://scikit-learn.org/stable/about.html

Proprietary Tools

The market for proprietary AI tools is massive – and it is likely to grow for the long haul. To get a sense of this, look at DataRobot. In November 2020, the company announced a whopping $270 million investment, which was led by Altimeter Capital. The valuation was set at $2.7 billion.[8]

This deal was not a one-off either. There have been a myriad of mega rounds for AI tools companies.

So then, who are some of the leaders in the space? Of course, mega tech operators like Microsoft, Amazon, and Google have their own platforms, which are based on their cloud offerings. Here's a look:

- *Azure Machine Learning*: This is Microsoft's suite of AI products, which includes internal systems but also third-party applications like Databricks and Keras. They are not just for data scientists. Microsoft also has applications geared for citizen developers and business analysts.
- *Google AI Platform*: There is BigQuery ML for handling massive data sets and access to the TensorFlow platform. The system also has easy-to-use AI applications for those who are not data scientists. Another interesting part of Google's system is the use of Kubeflow, which greatly helps with the deployment of applications. Finally, there is a thriving community of developers and researchers, such as with the AI Hub and Kaggle.
- *Amazon SageMaker*: The company has a huge advantage with its AWS (Amazon Web Services) platform. Thousands of customers already use this system to host their applications. This makes it easier to add AI capabilities. As for SageMaker, it is a full-blown system. "At Freshworks, we use AWS Sagemaker for distributed model training and workflow orchestration," said STS Prasad, who is the Freshworks EVP of Engineering. "Sagemaker enables us to train thousands of machine learning models in parallel for different customers and verticals."[9]

So now let's take a look at some of the other proprietary systems:

DataRobot

[8] www.datarobot.com/news/press/datarobot-announces-206-million-series-e-funding-round/

[9] From the author's interview with STS Prasad on June 7, 2020.

The company is one of the leaders in the AutoML space, which uses drag-and-drop and low-code approaches. DataRobot also handles all the key steps of model development and deployment (the program includes ten steps). The system can be used in the cloud or for on-premise environments and is integrated with AWS, Microsoft Azure, and Google Cloud.

According to research from Forrester, DataRobot demonstrated an ROI (Return on Investment) of 514% over a three-year period and a payback of less than three months. Note that the building and optimization of models took an average of 2.5 weeks, versus 13.5 weeks using other approaches.[10]

SAS

SAS is one of the early players in analytics and machine learning platforms. The company was founded in 1976 at North Carolina State University because there was a need for the USDA to analyze complex agricultural data. But this research would lead to opportunities across many other industries.

In the 1980s, the company revamped the software so it could run on any hardware or operating system. This certainly helped spur the growth. In fact, this strategy continues to be a key to the company's success today.

In terms of the AI capabilities, there was a release of SAS Viya in 2016. This allowed for much easier model creation and used a cloud-native architecture.

"Viya embeds key capabilities like machine learning, deep learning, computer vision, natural language processing (with conversational AI), forecasting, and optimization within the software so users have a seamless experience leveraging analytics capabilities to gain actionable insights and meet business needs," said Gavin Day, who is the Senior Vice President of Technology at SAS. "Viya also automates the process of data cleansing, data transformations, feature engineering, algorithm matching, model training, and ongoing governance, making it easier for both business analysts and data scientists to use SAS software. Results from queries are explained through natural language processing so data analyses are communicated in plain, easy to understand business terms."[11]

Then in 2019 the company announced its plan to invest $1 billion in AI technologies for a three-year period. This would include new products but also education and expert services.

Currently, SAS has close to 14,000 employees and 83,000 customer sites across the world. The company counts 91 of the top 100 companies on the 2019 Fortune 500 list as clients.

[10] https://3gp10c1vpy442j63me73gy3s-wpengine.netdna-ssl.com/wp-content/uploads/2020/05/ROI_Infographic_v.4.0.pdf

[11] From the author's interview with Gavin Day on June 15, 2020.

MathWorks

As you learned in Chapter 3, MathWorks is one of the pioneers of the analytics and machine learning tool markets. The company has two main products, MATLAB and Simullink. However, in terms of AI, MATLAB is really the one that is important to cover.

No doubt, an advantage for MathWorks is broad experience. This means that MATLAB has become a solid offering. It has a global ecosystem of users and partners. The system is also available across many platforms like on the edge, on-premise, and the cloud. You can even convert the algorithms to C/C++, HDL, or CUDA to be embedded in a processor for FPGA/ASIC.

Alteryx

Among the AI platform providers, Alteryx is the only pure-play operator that is publicly traded. The company launched its IPO in March 2017, with the stock price at $14. Since then, the shares have soared to $178 and the market cap has reached $12 billion.

Then again, Alteryx has been growing at a rapid pace. In the first quarter of 2020, the revenues jumped by 43% to $108.8 million and the company added 356 customers, including 12 of the Global 2000. In all, there are over 6,400 customers across the globe.[12]

According to Alteryx CEO and co-founder Dean Stoecker: "Competing, let alone winning in this data-driven world requires global enterprises to either disrupt themselves or be disrupted by others. It requires reimagining themselves, wherein data is valued as an asset and analytics as a prowess. This is not achieved by leveraging incumbent technologies and existing processes that made them great in the first place. And it cannot, in our view, be achieved by advocating analytics to only the trained statisticians working on edge cases, even with the best AI and ML capabilities. It can only effectively be achieved by harnessing the networking effects of people, data, and technologies, which allow companies to build a culture of data science and analytics that drives value across all functional areas of the organization. We believe this is best achieved with a human-centered platform that is code-free and code-friendly that liberates thinking, enables creativity, and analytics to address hundreds of use cases in every organization."[13]

The focus of Alteryx is to make its technology accessible by anyone. The system comes with hundreds of easy-to-use modules to build AI applications and models. There is also a robust data system, which ingests data sets from any source.

[12] https://investor.alteryx.com/news-and-events/press-releases/press-release-details/2020/Alteryx-Announces-First-Quarter-2020-Financial-Results/default.aspx

[13] www.fool.com/earnings/call-transcripts/2020/02/13/alteryx-inc-ayx-q4-2019-earnings-call-transcript.aspx

The company has also been smart with its acquisitions. To this end, it purchased ClearStory Data (automation of unstructured data) and Feature Labs (automation for feature engineering).

Dataiku

In 2013, the founders of Dataiku launched the company because they saw that using data was a strategic imperative. But the problem was that the available tools were simply not good enough.[14]

Its vision was certainly on target and the growth was immediate. As of now, hundreds of customers use Dataiku's technology to help with churn, fraud, predictive maintenance, and supply chain optimization.

In late 2019, the company announced a Series C funding round for $101 million, led by ICONIQ Capital. Other investors included Alven Capital, Battery Ventures, Dawn Capital, and FirstMark Capital.[15]

The Dataiku platform is called the Data Science Studio (DSS). A critical focus of the system is on allowing for teamwork and collaboration. As you've seen in this chapter, the model creation process involves numerous people across an organization, all with differing skillsets.

One of the company's marquee customers is BGL BNP Paribas, a global banking institution. The company uses the Dataiku platform to manage its fraud detection efforts. This is an area that requires ongoing attention, as security threats are constantly evolving. There is also the need to comply with changing regulations.[16]

While the company had its own set of machine language algorithms, they were difficult to update. So the company installed DDS, which took about eight weeks. This was also done with the necessary governance requirements.

However, the use of DDS was not limited to fraud detection. The system led to other projects to create more value.

Databricks

As you learned in Chapter 5, Databricks started as a project at UC Berkeley in 2009 and has since gone on to become a top player in the AI platform business. At the core of this is a sophisticated data infrastructure for all the steps in the model creation process. The system is based on Apache Spark, which is popular with machine learning. But Databricks is a contributor to other open source projects like Koalas, Delata, and MLFlow. And one of the most recent projects is Delta Lake, which helps clean up existing data lakes.

[14] www.dataiku.com/stories/the-dataiku-story/
[15] https://pages.dataiku.com/101million-series-c
[16] www.dataiku.com/stories/bgl-bnp/

The Databricks platform is cloud-based and available on AWS and Azure. Here are some of the use cases for the technology:

- It improves efficiencies for oil and gas companies. The machine learning is able to better discover and extract energy sources and help minimize downtime.
- Large pharmaceutical companies use Databricks to accelerate the drug development process. Databricks also quickly added COVID-19 data sets to the platform.
- Retailers are using the system to allow for personalized shopping experiences, such as with improved recommendations, pricing, and promotions.

An advantage of Databricks is that it allows a data scientist to use their preferred AI frameworks and libraries to interact with the underlying data. Then the model can be moved into production with a click. With a common UI, there are better feedback loops and collaboration.

Growth has been particularly strong for the company. In 2019, revenues exceed $200 million, up from $100 million on a year-over-year basis.[17] Customers include Hotels.com, Shell, Expedia, Regeneron, and Comcast.

KNIME

The roots of KNIME go back to 2004 when a team from the University of Konstanz created a platform to make it easier to manage and process data. The leader was Michael Berthold, who had a background in Silicon Valley. When the KNIME system was launched in 2006, there was much demand from the pharma industry. But over the years, the company has been successful in expanding across a myriad of verticals.

Consider that the KNIME Analytics Platform is open source but the KNIME Server is a proprietary solution that provides enterprise-level features like automation and collaboration. This combination is quite powerful, supporting the complete data science journey.

The KNIME open source system has a modern UI, which allows the user to drag-and-drop items on a canvas. This is much more intuitive than the typical script-based approach and allows for higher productivity.

[17] www.cnbc.com/2020/06/16/databricks-prepared-for-recession-with-fund-raising-real-estate-cuts.html

Other functions of KNIME include

- *Integrated deployment*: This is a workflow system that helps to minimize errors when deploying models. There are also a range of deployment options from a web-based dashboard and REST API service.
- *Data*: KNIME works with the most common data systems, such as SQL, NoSQL, cloud-based platforms, and on-premise databases.
- *Data wrangling*: You can use a variety of approaches like joining, concatenation, filtering, aggregations, and normalization.
- *Guide automation*: The whole data cycle can be automated, such as with ingesting, preprocessing, dimensionality reduction, outlier detection, and so on.
- *Model monitoring*: This handles issues with data drift or data jumps, which can distort models that have been deployed.

In-House AI Systems

The innovation with open source and proprietary AI systems continues at a rapid pace. But they all have their drawbacks. Let's face it, the AI process is highly complex and nuanced, and this is why some companies build their own solution. True, it may be expensive and time-consuming. But as AI becomes more strategic, an in-house system could wind up being a strong asset.

An example of a company that has gone down this route is Freshworks. Founded in 2010, the company operates a cloud platform that helps with customer engagement.

In late 2019, Freshworks raised $150 million from Sequoia Capital, CapitalG (which is affiliated with Google), and Accel at a valuation of $3.5 billion. The company has more 40,000 paying customers like Cisco, Hugo Boss, and Honda.[18]

Even though Freshworks has its own in-house system, the company still uses a variety of third-party tools. "At Freshworks, we use AWS Sagemaker for distributed model training and workflow orchestration," said STS Prasad, who is the EVP of Engineering at the company. "Sagemaker enables us to train thousands of machine learning models in parallel for different customers and

[18] https://techcrunch.com/2019/11/13/freshworks-raises-150m-series-h-on-3-5b-valuation/

verticals. We also use Apache Spark, Hadoop, Sci-kit, and R libraries for the most common machine learning algorithms. For deep learning, we use Keras and PyTorch frameworks and pre-trained deep learning models such as BERT and ELMo for embedding text."

Then what about the in-house AI systems? The focus is primarily on configurability and productivity for data scientists. For the most part, the goal is to find ways to get models productized as quickly as possible.

And yes, Freshworks used this to create an AI system called Freddy AI, which its customers can use. "Freddy AI continuously learns from all customer interactions across marketing, sales, and support," said Prasad. "The Freddy chatbot enables businesses to acquire, engage, and support customers without the need for manual intervention. Freddy AI learns from ticket data in our customer engagement product, Freshdesk, to help automate repetitive tasks, provide self-service for routine questions, and allow for contactless resolution of service requests."

Freddy AI operates on top of the Freshworks master customer record, consolidating data across the Freshworks suite of products to give organizations a 360-degree view of their customers. There are also heavy investments in conversational AI like agent response suggestions and type-assist features (similar to Gmail's smart complete)

"Every business is different, and each customer requires a certain level of configurability in order to make AI/ML features work to meet their needs," said Prasad. "Because of this, every Freddy AI feature comes with a standard set of admin configuration options that can be used to tailor its features to suit the unique needs of an organization. Another lesson is that our customers aren't just looking for the newest or trending AI capabilities, they're looking for features and actionable insights that drive business transformation. AI/ML initiatives have to be ROI positive for continued adoption."

Correlation Vs. Causation

There are a variety of rules of thumb when creating models. In this chapter, you've taken a look at some of them, such as dividing the data set into different sections.

Now these rules of thumb may not always be correct. However, they can still be quite helpful and streamline the AI process.

But there is another very important rule of thumb to keep in mind, and you do not have to be a data scientist to use it: Correlation is not causation. There are actually some comical examples of this. Just look at Tylervigen.com, which collects faulty correlations. Here are a few examples:

- The number of people who drowned by falling into a pool correlates with films Nicolas Cage appeared in.
- Per capita cheese consumption correlates with the number of people who died by becoming tangled in their bedsheets.
- The divorce rate in Maine correlates with per capita consumption of margarine.

It's kind of crazy. To this end, there is a concept known as patternicity, which means we have a tendency to detect patterns in meaningless noise!

The fact is that finding causation is extremely difficult and time-consuming. It usually involves extensive studies. For example, in medical research, there is research that involves randomized samples and control groups to see if certain conditions lead to a particular disease. One of the most famous examples of this involved the studies on smoking and lung cancer during the 1960s.

Something else about when using AI models: There are oftentimes interesting relationships that seem counterintuitive and yet are still spot on. These types of insights can certainly be a big driver for a business. Here are some examples:[19]

- People who completed a loan form correctly and without spelling errors had better risk profiles. This was less so for those who used all lower case.
- According to data from Xerox and other large companies, those employees who used Chrome and Firefox performed better. Why so? The theory is that these employees were willing to experiment and try new things to help improve their productivity.

Note What's the difference between machine learning and artificial intelligence? If it's written in Python, it's probably machine learning. If it's written in PowerPoint, it's probably AI. In other words, when it comes to AI, there remains quite a bit of hype!

[19] https://blogs.scientificamerican.com/guest-blog/9-bizarre-and-surprising-insights-from-data-science/

Ending a Project

The best practices explained in this chapter will go a long way to improving the odds of success of an AI project. But unfortunately, there will likely be cases where a project may not be viable. This may come after much investment and iterations.

Yet this should not be seen as a failure. It is just an inevitable part of the process. AI is generally based on massive data sets and probabilities. As a result, it's impossible to guarantee a good outcome.

But a failed project should have a postmortem. Take an honest look at why it did not work out and what lessons were learned. This will certainly be essential for the next project.

Conclusion

In this chapter, you looked look at the steps and best practices for creating models. This is a challenging process but also engaging. You may learn an insight that could move the needle for your business. On the other hand, a project may just wind up being a failure. But with the process, you will continue to build your AI muscles.

OK then, regarding the next chapter, we will cover the deployment of AI.

Key Takeaways

- Model selection should start with experimentation. There are hundreds of algorithms available.
- But before selecting a model, you need to look at the quality of the data set, the deployment options, maintenance requirements, and any legal issues.
- It's usually best to start with simple models.
- Custom AI models are certainly powerful. But they usually require PhDs to create. Publicly available models are often sufficient for many business applications.
- An ensemble model is when more than one AI model is combined. This can lead to improved accuracy scores, although there will be increased complexity.
- An initial step in the training of an AI model is to randomize the data. This will help reduce the incidence of false patterns.

- The next step is to divide the data set into at least two parts. One of the segments will be for testing (which will have 70% to 80% of the information), such as to find the parameters and features of the model. After this, there will be a validation data set. This will be used to get a sense of how the model works in the real world. There will also be a look at issues like gaps in the data, underfitting, and overfitting.
- Model building is complex and requires technical skills. But there are some web-based systems that allow just about anyone to develop their own models. Examples include Teachable Machine from Google and the Machine Learning Playground.
- Feature engineering is about identifying the variables that are the best predictors for a model. This often means having a good understanding of a domain. For example, an expert should be able to come up with the typical factors that help explain certain patterns and systems.
- Sophisticated deep learning can be used to find features that are often not detectible by people. But this approach is far from perfect. There should still be a human in the loop to check the results.
- A parameter is a variable that is internal to the model and whose value is determined from the underlying data.
- A hyperparameter is a value that is external to the model and cannot be estimated from the data. A data scientist will specify the hyperparameter, using some type of rule of thumb. This is known as hyperparameter tuning. There are machine learning systems that can also automate this.
- Accuracy is the measure of the correct predictions divided by the total predictions. But there are many other ways to get a sense of the accuracy of a model. Some of these include metrics like precision, recall and the F1 score. They account for the false positives and false negatives in the model.
- There are a large number of AI tools. Some are open source, which means they are freely available. And others are proprietary and require a license or subscription. Both solutions have pros and cons. It all depends on what you want to accomplish with your AI efforts.

- Python is the most popular programming language for AI. It is relatively easy to learn and there is an extensive ecosystem of third-party add-ons like NumPy and Pandas.
- In some cases, a company will build in-house AI tools. This option is expensive but can be worth the effort, so long as the focus is on strategic capabilities.
- A key principle in AI is that "correlation is not causation." If you want to find the real cause of something, you must perform in-depth studies and analysis.
- Failure is common with AI models, even when there is a solid process. This is ok. But it is important to do a postmortem to see what happened.