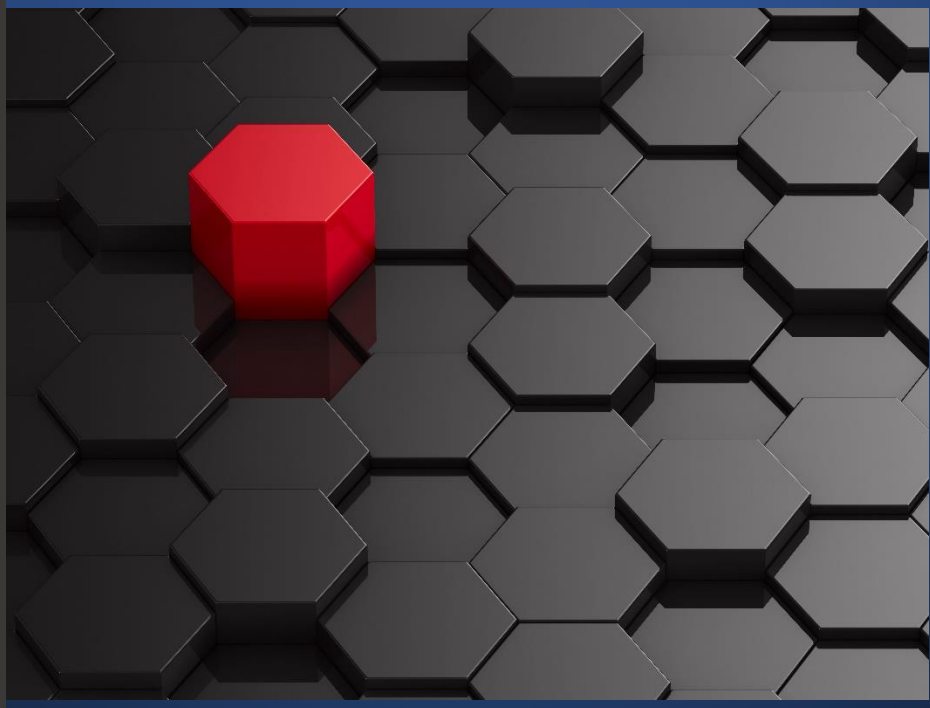
PATTERN
COMPUTER®



WHAT MAKES PATTERN COMPUTER DIFFERENT?

Ty Carlson

Pattern Computer Inc.

38 Yew Lane

Friday Harbor, WA 98250

What Makes Pattern Computer Different?

Nearly all Artificial Intelligence (AI) and Machine Learning (ML) companies use a collection of known techniques to be able to recognize specific things, such as cars, road signs, faces and songs. Similar methods allow prediction of outcomes such as sports scores, insurance rates, and risk calculations, based upon known historical data. By contrast, Pattern Computer finds things that have not been found before. We find patterns in datasets that are generally too large or computationally complex for patterns to be discovered. We find what others cannot. That may be a pattern behind a specific cancer, a collection of sensor readings indicating premature part failure, or the operational patterns causing flight departure delays. Once we have discovered the key features of the pattern, we create an accurate mathematical model of the response.

Machine Learning – how we got here.

In the 1970s, computer researchers wanted to model aspects of the human brain, particularly how a collection of inputs passing through a network of decision points (neurons) could lead to an informed outcome based upon the varying nature of the inputs. The computational algorithm was referred to as a neural network. They were able to produce some insights, but the amount of computational power required to do useful work exceeded the capability of existing systems, and their work was shelved.

Periodically, work on neural networks would be revived, but the lack of reasonable computational capability necessary to handle the exponential nature of the algorithms continued to be a barrier to further research. Finally, in the mid-2000s, the evolutionary development of graphical processing units (GPUs) - first as a computational accelerator and later as a general-purpose graphical processing unit (GPGPU) - created the ideal computational platforms for neural networks.

Why? A neural network looks at a series of decision points (nodes), which are contained within the sequential layers of the network and performs similar computational steps[1] at each node. This design is well-suited to GPUs because they are massively parallel and can process hundreds of thousands of computations simultaneously across the nodes. The iterative improvement of GPUs and the games which took advantage of this capability drove the popularity of gaming platforms which helped to fund the R&D costs to create a virtuous cycle of development.

Nvidia noticed this capability, and developed programming interfaces for its GPUs, helping to usher in the GPGPU with their CUDA programming environment. Development of neural networks accelerated during this time, with processing power doubling roughly every two years being pushed by Nvidia's release cadence. With that incredible computational growth and continued research work, neural networks evolved into a viable platform for machine learning.

The process of machine learning is a series of stages: training, validation, and testing. Each stage uses a different subset of the target dataset. The training set, typically 50-90%[2 of the dataset], is used to learn network parameters which allow it to better classify samples in the dataset. An example task in the context of self-driving cars is discriminating between a yield sign, a speed limit sign, and a 'slow' sign. As

[1] Using SIMD (single instruction, multiple data) operations
[2] This is often very dependent on the number of samples (observations) in the dataset.

part of the training process the neural network is provided with annotated images of each road sign being processed by the algorithm. After training is completed, the neural network acts as a model for recognizing those three types of road signs.

In the next stage, validation data is run thru the model and the results are used to as checks to determine the accuracy of the model. The developer or data scientist then likely adjusts the configuration settings (hyperparameters) for the neural network (such as the number of layers in the network, or the number of nodes at each layer) to see if they can increase the accuracy of the results with the validation set. The risk of such tuning however is that the parameters can become too narrowly focused on the validation dataset (called *overfitting*) and no longer produce a generalized model representing the relationships in the dataset.

Once a final model is proposed, the model is given the test dataset to check the model's accuracy on the held-out data, and to check for overfitting (where the training set and validation set accuracy results are much higher than that of the test set).

It is important to note that neural networks do not encompass the entire world of machine learning techniques. While Pattern Computer uses neural networks, both as core engines for certain critical work items and as complementary tools for other work items, Pattern Computer also uses other methods and techniques to discover patterns within high dimensional datasets. The goal is to reduce the complexity of the dataset, while maintaining an understanding of the relationships between the important features within the dataset.

How is Pattern Discovery different?

The success and popularity of pattern *recognition* has directed a significant amount of energy and investment into the space of neural networks, including very elegant and refined variations of neural networks based on convolutional neural networks, recurrent neural networks, and variations including back propagation. To truly have an informed understanding of how to apply neural networks and how the network works, it is useful to have a deeper understanding of the mathematics behind neural networks and the nature of the datasets you wish to investigate. At Pattern Computer, depending on those characteristics, we may choose to use a specially tuned neural network with a pre-processed dataset that has been reduced to the significant related elements. The pre-processing allows us to:

- Do minor transforms on the data objects
- Only include the data of interest
- Reduce size of the dataset[3].

Using these techniques, we can observe clustering of the data objects and understand the relationships between similar things that have the desired characteristics. From this basis, we can build models of relationships, to gain new understandings of the system represented by the dataset. The patterns thus discovered become a model for investigation and interpretation by subject matter experts in that field. For example, when investigating the related gene mutations in a specific type of lung cancer, a subject matter expert can quickly understand related biological mechanisms impacted, or genomic pathways affected.

[3] This is important for performance reasons as typical neural networks are $O(n^2)$

Alternatively, we may choose to use a different mathematical method, where we use acyclic digraphs to build a model of the data represented in the dataset. This is a complex and very detailed task, which we perform at scale. We then perform an analysis on that model to understand the nature of the relationships within the data to identify the most significant features in dataset associated with a specific outcome. The number of features is determined by the nature of the data itself, rather than having the system set an artificial limit. This contrasts with methods typically used by other companies or researchers investigating datasets where the dimension is artificially constrained by the researchers (due to complexity), or by the data scientist due to the computational complexity, time and often exponential cost of increasing the upper limit of the feature dimensions.

Not only do we identify the most significant sets of features associated with a specific outcome, but we also note the relationships of the data features within the model and the structure of the data itself. Each of these relationships may be informative. Some of these relationships are tightly embedded within the data, while others can be viewed topographically. Pattern Computer has spent over 3 years to apply mathematicians, bioinformaticians, microbiologists, physicists and control theorists towards develop these algorithms. We encountered many limitations in the mathematics that needed to be solved or addressed using creative means. When we felt we had the mathematics solution, the next step was to engineer the algorithms to run at scale.

Where the general math was simple, the details were complex, and vice-versa. The goal of Pattern Computer is to discover patterns in high-dimensional datasets – with one such target being cancer in humans, where there may be up to 24,000 genes of interest for each patient. If you are working in the space of colorectal cancer, the GECCO dataset includes more than 39 million single nucleotide polymorphisms (SNPs – “snips”) per patient. The size of the data gets extremely large very quickly. To handle these problems, we needed to have significant flexibility in our design to allow these issues to be solved using our size-reduction algorithms. We implemented a flexible architecture to ingest the datasets and allow the acyclic digraph structures to exist in memory on the same system (for a given stage of processing) where possible.

The design of the systems architecture …

complements the algorithm design …

which reflects the mathematics …

based on the nature of the dataset(s) …

All these components must work well together to produce the desired capability. Do we use existing techniques that have been published and may be well-understood? Yes (of course). Do we use novel techniques which have not been published and may not be well understood? Yes. Absolutely. Where prudent, we have applied for patents. Where little is known or published in the more challenging sections, we have reserved those as trade secrets.

In some cases, where Pattern Computer may be working on a problem with mixed data and tabular images, we have the option to use a series of different techniques, combining our innovative approach to genomics using a neural network-based discovery engine we call “Sonar” and combine that approach with the high-dimensional reduction discovery engine we call “Lennard Island”. Using these two approaches we use different computer architectures in a distributed system to develop the best approach to discovering the full set of patterns.

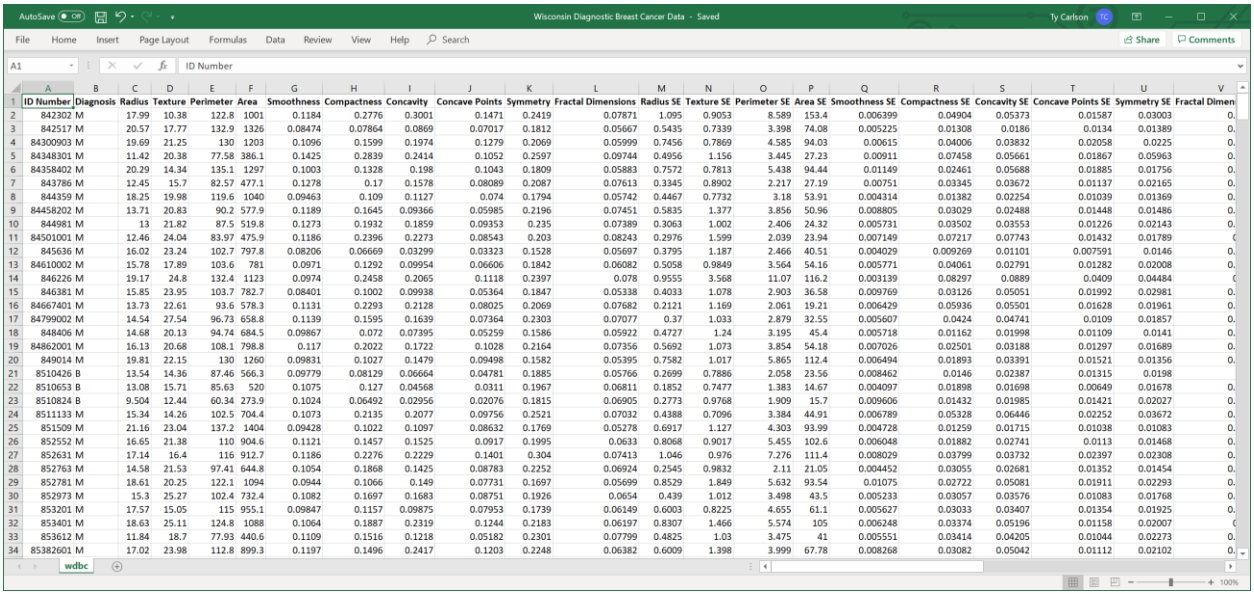
To answer the question of how Pattern Computer does the discovery, our answer is that we use a series of mathematical techniques applied with insight gained from the nature of the dataset. We have developed very scalable applied mathematics with insights into the nature of acyclic digraphs. These algorithms were written with an awareness of some of the rules and caveats required when working with high dimensional structures. Our very experienced mathematicians applied keen observations about the nature of these types of structures and had the foresight to avoid getting caught in blind alleys. The result is a flexible overall solution where different mathematical approaches may combine to take the best characteristics of each solution and share that information to provide hints and insight into the specific and combinatorial relationships within the data. In some cases, we transform the data to ingest and process the entire dataset. It has taken over 3 years to optimize the mathematics, the algorithms, implement distributed scaling and systems acceleration in an iterative process. With each test of large datasets, we learned more about the specific performance characteristics of the algorithms, and how the performance is affected by the ratios of different datatypes contained in the dataset. It truly has been a collaborative journey of exploration, development, testing and performance analysis.

We are different because:

- We are doing pattern *discovery*, not pattern recognition.
- We identify a ranked list of the feature sets associated with a specific outcome. You know what features are driving the specific outcome – this is in sharp contrast to typical implementations of neural networks.
- We build mathematical model of the nature of the relationship between the features in the feature set that maps to the specific outcome.
- We can directly map the ranked feature set to the set of observations supported by that feature set which associate with the specific outcome.
- We can identify the outlier features which have no apparent influence on the outcome.
- The “AI” that is produced through Pattern Computer is easy to understand and is “safe ai” – not hidden in a black box.
- Our focus is understanding the nature of the mathematics and designing flexible algorithms which avoid the shortcomings of some techniques in transforming and processing large datasets. That approach allows us to accomplish the pattern discovery work.
- Our Chief Architect designed the immediate predecessor to AutoCAD, he still counts bits and bytes. He knows how to efficiently work with very large matrix operations, manage memory, and implement high-performance parallel solutions.
- We are not a just a team of software engineers implementing known methods. We have experienced researchers in science, applied mathematics, microbiology, bioinformatics, applied physics, computer science and digital logic design who actively collaborate to solve novel problems.
- We have an active group working on understanding the underlying nature of neural networks, referred to as Explainable AI (XAI). We have already developed insights on the nature of the layers within the network, as well as methods to optimize the neural network for size and performance considerations.
- Pattern Computer is not only looking at tabular data but combining tabular and imaging information as input to the pattern discovery process.

A Short Example

Let's look at a very small example in biology using a real, public dataset – the Wisconsin Diagnostic Breast Cancer dataset (WDBC) from November 1995. This is a dataset of 965 breast cancer patients with 30 inputs based on features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image (http://www.cs.wisc.edu/~street/images/).



Wisconsin Diagnostic Breast Cancer Data - Saved

ID Number	Diagnosis	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Concavity	Concave Points	Symmetry	Fractal Dimensions	Radius SE	Texture SE	Perimeter SE	Area SE	Smoothness SE	Compactness SE	Concavity SE	Concave Points SE	Symmetry SE	Fractal Dimen
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4	0.006399	0.04904	0.05373	0.01587	0.03003	0.
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.005225	0.01308	0.0186	0.0134	0.01389	0.
84300903	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.00615	0.04006	0.03832	0.02058	0.0225	0.
84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23	0.00911	0.07458	0.05661	0.01867	0.05963	0.
84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.01149	0.02461	0.05688	0.01885	0.01756	0.
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217	27.19	0.00751	0.03345	0.03672	0.01137	0.02165	0.
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18	53.91	0.004314	0.01382	0.02254	0.01039	0.01369	0.
84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	0.5835	1.377	3.856	50.96	0.008805	0.03029	0.02488	0.01448	0.01486	0.
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32	0.005731	0.03502	0.03553	0.01226	0.02143	0.
84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039	23.94	0.007149	0.07217	0.07743	0.01432	0.01789	0
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466	40.51	0.004029	0.009269	0.01101	0.007591	0.0146	0.
84610002	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564	54.16	0.005771	0.04061	0.02791	0.01282	0.02008	0.
846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2	0.003139	0.08297	0.0889	0.0409	0.04484	0
846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58	0.009769	0.03126	0.05051	0.01992	0.02981	0.
84667401	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21	0.006429	0.05936	0.05501	0.01628	0.01961	0.
84799002	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879	32.55	0.005607	0.0424	0.04741	0.0109	0.01857	0.
848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4	0.005718	0.01162	0.01998	0.01109	0.0141	0.
84862001	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854	54.18	0.007026	0.02501	0.03188	0.01297	0.01689	0.
849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.017	5.865	112.4	0.006494	0.01893	0.03391	0.01521	0.01356	0.
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058	23.56	0.008462	0.0146	0.02387	0.01315	0.0198	
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383	14.67	0.004097	0.01898	0.01698	0.00649	0.01678	0.
8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7	0.009606	0.01432	0.01985	0.01421	0.02027	0.
8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4388	0.7096	3.384	44.91	0.006789	0.05328	0.06446	0.02252	0.03672	0.
851509	M	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.05278	0.6917	1.127	4.303	93.99	0.004728	0.01259	0.01715	0.01038	0.01083	0.
852552	M	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995	0.0633	0.8068	0.9017	5.455	102.6	0.006048	0.01882	0.02741	0.0113	0.01468	0.
852631	M	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304	0.07413	1.046	0.976	7.276	111.4	0.008029	0.03799	0.03732	0.02397	0.02308	0.
852763	M	14.58	21.53	97.41	644.8	0.1054	0.1868	0.1425	0.08783	0.2252	0.06924	0.2545	0.9832	2.11	21.05	0.004452	0.03055	0.02681	0.01352	0.01454	0.
852781	M	18.61	20.25	122.1	1094	0.0944	0.1066	0.149	0.07731	0.1697	0.05699	0.8529	1.849	5.632	93.54	0.01075	0.02722	0.05081	0.01911	0.02293	0.
852973	M	15.3	25.27	102.4	732.4	0.1082	0.1697	0.1683	0.08751	0.1926	0.0654	0.439	1.012	3.498	43.5	0.005233	0.03057	0.03576	0.01083	0.01768	0.
853201	M	17.57	15.05	115	955.1	0.09847	0.1157	0.09875	0.07953	0.1739	0.06149	0.6003	0.8225	4.655	61.1	0.005627	0.03033	0.03407	0.01354	0.01925	0.
853401	M	18.63	25.11	124.8	1088	0.1064	0.1887	0.2319	0.1244	0.2183	0.06197	0.8307	1.466	5.574	105	0.006248	0.03374	0.05196	0.01158	0.02007	0
853612	M	11.84	18.7	77.93	440.6	0.1109	0.1516	0.1218	0.05182	0.2301	0.07799	0.4825	1.03	3.475	41	0.005551	0.03414	0.04205	0.01044	0.02273	0.
85382601	M	17.02	23.98	112.8	899.3	0.1197	0.1496	0.2417	0.1203	0.2248	0.06382	0.6009	1.398	3.999	67.78	0.008268	0.03082	0.05042	0.01112	0.02102	0.

Figure 1: Wisconsin Diagnostic Breast Cancer Dataset

The results are based on predicting whether the cancer is benign or malignant (Column B in Figure 1). This dataset has been posted on a machine learning competition site (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data), where the top predictive results have ranged between 97.5% and 100.0%, using various algorithms including neural networks or support vector machines. While Pattern Computer can achieve similar results using neural networks as well as our internal proprietary methods, what is different about Pattern's solution is that *can identify the critical features (columns) that produce the result. Additionally,* we can extract the mathematical relationship between those features to predict the results in other target data.

In the typical case of the neural network approach, held-out data is provided to the model created by the neural network and the diagnosis based on patient tumor data is returned – whether the patient's tumor is benign or malignant. The next question it asks is, "Why? What are the factors that make the model think the tumor is malignant?"

In general, neural networks will not tell you what the important factors are in the result, or how the model is constructed. Of course, knowing that would be extremely useful, particularly if the source data is difficult or expensive to collect and process. If a researcher knew the key observations necessary to determine if a tumor is malignant or benign, they would gain valuable insights on how to understand and identify malignant tumors. After running the Wisconsin Diagnostic Breast Cancer dataset through

our Pattern Discovery Engine, we know that the two most critical features to distinguish between malignant and benign tumors are:

- Area Worst[4] (column Z) and
- Concave Points Worst[5] (column AD)

These two values can predict the balanced accuracy (predicted vs. actual) of 93.9% / 93.8% (benign vs malignant).

Being able to have such high accuracy with only two features provides great simplicity and parsimony in understanding the key factors associated with the identification of the nature of the tumor. Having just two features also reduces the chance of creating an incorrect general model due to overfitting. With this model a data scientist can place the equation below into an Excel spreadsheet alongside the data and can immediately see the modeled prediction. Finally, given a clean dataset such as Wisconsin Diagnostic Breast Cancer, we can produce these results in less than 30 minutes.

Moreover, we know the mathematical model that produces these results.

```
If (((ln(0.827546 * area_worst)) + (-0.049885 *
concave_points_worst*area_worst))) >= 1.450000 then the tumor is benign.
```

The ability to build and understand a mathematical model which accurately predicts the result is a powerful tool and fundamentally different from generic "artificial intelligence" solutions. With this model we can clearly see both additive and multiplicative contributors to the outcome and the relationship of the different features to one another. Instead of needing the 30 values to produce the predicted response, we can identify the most important features contributing to the specific outcome.

We can also produce more complex result sets, with higher accuracy using more features, such as the highest-ranked third-order feature set given by the Pattern Discovery Engine:

- Perimeter Worst[6]
- Texture Worst[7]
- Concave Points Mean[8]

In this case the mathematical model is:

```
If ((((sqrt(concave_points_mean)) + (sqrt(0.019171 *
perimeter_worst)) + (-0.052156 * perimeter_worst/texture_worst)))
< 1.425) then tumor is benign.
```

[4] Area Worst is the mean of the three largest values of the area of the cell nucleus.
[5] Concave Points Worst is the mean of the three largest values of the number of concave portions of the contour of the cell nucleus.
[6] Perimeter Worst is the mean of the three largest values of the perimeter of the cell nucleus.
[7] Texture Worst is the mean of the three largest values of the standard deviation of gray-scale values of the digitized image of the cell nucleus
[8] Concave Points Mean is the mean of the number of concave portions of the contour of the cell nucleus.

In this case, we see **96.6% / 95.8%** accuracy (benign vs. malignant), using three features instead of two.

In summary

The ability to identify which features are specifically associated with a given outcome, or set of outcomes, plus the ability to build and know the mathematical model which represents the data and predicts those outcomes is a powerful and meaningful differentiator in the machine learning space. You cannot get these insights and answers by simply running a neural network. Furthermore, the Pattern Discovery Engine is a very powerful and highly optimized solution capable of producing results in a matter of days, if not hours, versus the weeks or months it would take with neural networks.

- ✓ Yes, we discover previously unknown patterns.
- ✓ Yes, we produce mathematical models of the methods.
- ✓ Yes, this is different.
- ✓ Yes, it produces real results.

Welcome to Pattern Computer!