Artificial Intelligence and Machine Learning are two of the hottest buzzwords in the field of Information Technology. Artificial Intelligence is the broader term and is defined by Investopedia this way:

*Artificial intelligence is a term for simulated intelligence in machines. These machines are programmed to “think” like a human and mimic the way a person acts. The ideal characteristic of artificial intelligence is its ability to rationalize and take actions that have the best chance of achieving a specific goal, although the term can be applied to any machine that exhibits traits associated with a human mind, such as learning and solving problems.*

Artificial Intelligence (AI) is behind the growing popularity of the *Virtual Digital Assistant* (VDA) as popularized by *Google Home, Siri, Cortana* and *Alexa* and used by consumers to answer questions and automate everyday tasks. Business are increasingly using VDAs for sales, marketing and customer service applications as well.

For example, Bank of America’s *Erica* serviced 3.5 million users and 11 million transactions within three months of its launch, helping banking customers to check their account balances, monitor transactions, access account numbers and check credit scores. Allstate’s VDA, referred to as *Abie*, helps agents to quote business insurance products quickly and accurately, while reducing call center traffic and providing instant answers to policy questions during the quotation process.

According to market research firm Tractica, active consumer VDA users will grow from 390 million in 2015 to 1.8 billion worldwide by the end of 2021.

During the same period, active enterprise VDA users will rise from 155 million in 2015 to 843 million by 2021.

The market intelligence firm forecasts total VDA revenue will grow from $1.6 billion in 2015 to $15.8 billion in 2021.
Machine Learning (ML) is a subset of artificial intelligence and is the enabling technology behind the rapidly growing field of predictive analytics. Machine learning uses sophisticated algorithms that allow computers to recognize patterns from current and historical data, learn from those patterns and then make predictions about future outcomes.
For example, those outcomes might be behaviors a customer is likely to exhibit during a shopping experience or anticipate possible changes in the stock market.

Predictive analytics help business leaders to understand and predict possible future occurrences by analyzing the past.

Internet-based applications of machine learning are becoming commonplace – events that appear in your Facebook feed, product recommendations made by Amazon, and movie suggestions presented in Netflix – they all make predictions based on data patterns analyzed by machine learning algorithms.

Predictive analytics has become a high-growth market opportunity. According to Statistics MRC, the Global Predictive Analytics market is expected to grow from $3.89 billion in 2016 to reach $14.95 billion by 2023 with a CAGR of 21.2%. Fueling this rapid growth of machine learning and predictive analytics are a multitude of business applications that cross virtually every industry sector.

Here are a few examples of machine learning in action:

**Recommendations Engine:** Suggestions to buyers for related products during the shopping and purchasing process to cross-sell and upsell the value of transactions.

**Fraud Detection:** Looking for patterns and behaviors that serve as markers for criminal or fraudulent behavior during the operation of online transactions.

**Personalized Marketing:** Segmenting and targeting marketing campaigns to match high-value buyers with high-probability purchases based on historical and demographic data analysis.

**Operational Efficiency:** The use of predictive models to forecast inventory levels and manage enterprise resources based on historical operating data.

**Dynamic Pricing:** Setting the optimum price levels for products and services by analyzing changing market conditions and consumer demand (e.g., airline ticket and hotel occupancy).

**Risk Reduction:** Using credit scores to assess the likelihood of defaulting on a purchase, evaluating the risk of insurance claims, or predicting the outcome of a collections process.

**Health Care Applications:** Using machine learning to predict the likelihood that patients will develop a chronic disease or how they will respond to a potential treatment plan.

**Insurance Applications:** Improve the process of underwriting, pricing, preventing fraudulent claims and optimizing marketing programs targeting business or consumer segments.

**Predictive Maintenance:** The use of machine learning and vehicle data to help predict which components might fail and recommend preventative actions.

**MEETING THE TEST DATA CHALLENGE FOR AI AND ML**

When developers and data science practitioners think about new applications for AI, ML and predictive analytics, they often think the bulk of the work will be in the development of the algorithms and how to code them. However, the biggest challenge is often on provisioning the data used to train, validate and test the model for accuracy and robustness. When perfecting a new algorithm for AI and ML applications, it helps to remember this simple rule of thumb:
The Accuracy of Algorithms used for AI and ML = High Quality Training and Test Data at Scale

How much training data is enough? To answer that question, simply ask yourself how accurate the results must be. The more data used to train and test the model, the better the learning process and the higher the accuracy of the results.

The greater the volume and variety of training data used, the more accurate and robust the model for predicting future outcomes will be. The challenge is this: How to provision a high volume of high-quality training data without spending an enormous amount of time collecting, labeling, classifying, cleaning, pruning, normalizing, and formatting the data with the help of domain experts who understand the data requirements.

Also, important to remember is the need for 3 different kinds of data during the development process: One dataset is needed to train the model, one dataset to validate the model and one dataset to test the model.

**DATA REQUIREMENTS FOR ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

**Training Dataset:** The sample of data used to train and evolve the accuracy of the model.

**Validation Dataset:** The sample of data used to provide an unbiased evaluation of the model.

**Test Dataset:** The sample of data used to provide a final test of the model prior to release.

These 3 datasets must be different to ensure the integrity of the model and how it will perform in real-world applications scenarios. That’s where GenRocket’s ability to generate high-volumes of data based on a predefined data model, data attributes and patterns of data variation is a perfect match for AI and ML application development. Once the domain expert specifies the data requirements, GenRocket’s real-time synthetic test data engine generates controlled and conditioned data at the rate of 10,000 rows per second. This allows developers and testers to create very large datasets on-demand for the separate purposes of training, validating and testing a machine learning application.
THE TECHNICAL CHALLENGE

• A global services organization developing a new Artificial Intelligence application needed a large dataset of customer data with given pattern of data to train the system.

• While there were many, many use cases, one use case example was where GenRocket generated 10 million rows of customer data, with a given pattern of:
  
  • 20% of the Customer Names should have either Jr / Sr. (60%) or Mr / Mrs (40%) with same customer details like address, but different Phone number and Unique Identification Number.
  
  • The unique identification number for each customer should have a unique alphanumeric character consisting of: <2 letters> <2 digits> <2 digits> <2 digits> <1 letter> Eg. AB 12 34 56 C

THE GENROCKET SOLUTION

• Generating data with complex business logic can be done easily with GenRocket by breaking it down into simple concerns.

• Linked Generators were used to design complex patterns of data where the percentage of data generated was controlled from multiple lists by using Generators called the WaitAmountGen and WaitAmountReferenceGen; these Generators referenced another Domain Attribute’s generated value.

• Linked Generators is a powerful GenRocket feature because linked Generators are able to directly reference each other within an Attribute as opposed to indirectly accessing another Domain’s Attribute to get its Generated value. Thus, linking Generators to each other within an Attribute provides the ability to generate complex conditioned data without the necessity of having to access another Attribute’s generated value. This also means that Attributes may reference other Attributes that generate complex data via linked Generators, thus yielding even more complex, conditioned data.

SETUP

• A New Domain was created using the New Domain option in the GenRocket web app and the appropriate Generators were added to each Attribute.
• The following image shows how the Customer name was Generated using Linked Generators.
• For Creating the Names with the given condition, an additional Attribute was created with the MultiWeightGen Generator, which assigned the value ‘1’ 80% of the time and the value ‘2’ 20% of the time. This was to ensure that 80% of customer data had the condition of the different first name, last name, address, city, zip code phone number and unique id and 20% of customer data had the condition of the same second name, address, city and zip code.
• By using Linked Generators the patterns and percentages of data generated from multiple lists were controlled and conditioned by referencing the MultiWeightGen Generator’s generated value. The following image shows the parameters set in the WaitAmountReferenceGen Generator for generating First name with Jr / Sr. (60%) and Mr / Mrs (40%)

• To ensure the same value was repeated for the data being generated for all the Attributes, a linked WaitAmountGen Generator was added and referenced to the value generated by the MultiWeightGen Generator. The following image shows the AddressGen being referenced to the MultiWeightGen for the waitAmount. This process was repeated for second name, address, city and zip code.
• Unique Identification Numbers like different countries’ identification numbers were generated by using the StringRegexGen Generator. The following image shows a sample of data generated by using StringRegexGen by converting the given condition <2 letters> <2 digits> <2 digits> <2 digits> <1 letter> to a regex `[A-Z]{2} \d{2} \d{2} \d{2} [A-Z]`.

```
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Preview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview - Loop Count:</td>
<td>25</td>
</tr>
<tr>
<td>gen1</td>
<td></td>
</tr>
<tr>
<td>UX 83 42 61 C</td>
<td></td>
</tr>
<tr>
<td>DW 12 45 98 B</td>
<td></td>
</tr>
<tr>
<td>KY 68 14 25 X</td>
<td></td>
</tr>
<tr>
<td>DM 59 39 95 Y</td>
<td></td>
</tr>
<tr>
<td>JK 71 91 70 M</td>
<td></td>
</tr>
<tr>
<td>CD 36 46 88 T</td>
<td></td>
</tr>
<tr>
<td>QF 78 10 79 I</td>
<td></td>
</tr>
<tr>
<td>VO 86 85 71 V</td>
<td></td>
</tr>
<tr>
<td>DS 16 51 73 M</td>
<td></td>
</tr>
</tbody>
</table>
```

• The Unique Identification Number was created by simply using StringRegexGen and adding the regex value in the regex parameter. The following image shows that the regex value was added to the regex parameter in the StringRegexGen.

```
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Preview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: gen1</td>
<td></td>
</tr>
<tr>
<td>Generator: StringRegexGen</td>
<td></td>
</tr>
<tr>
<td>regex: [A-Z]{2}\d{2}\d{2}\d{2}\d{2}[A-Z]</td>
<td></td>
</tr>
<tr>
<td>seed: Double Click to Add a Reference</td>
<td></td>
</tr>
<tr>
<td>Save Generator</td>
<td>Remove Generator</td>
</tr>
</tbody>
</table>
```
• The image below shows preview data generated in the GenRocket web app matching the required data criteria.

LARGE VOLUME DATA GENERATION

• Standard GenRocket data generation is between 10,000 to 15,000 rows of test data per second.

• For the use case 10 millions rows of data was taking over 16 minutes so the decision was made to use the GenRocket Partition Engine to speed up data generation.

• To generate the large volumes of data in delimited file format using the Partition Engine, the DelimitedPartitionReceiver was used. This Receiver outputs data in a delimited file format to one or more files parsed over multiple instances via the GenRocket Partition Engine. This allowed for huge amounts of data to be generated, in parallel, quickly. The generated files were then merged together into a single file using the PartitionFileMergeReceiver.

• In the DelimitedPartitionReceiver Parameters tab, the client defined the output directory in which the generated data was to be stored, the number of records generated per file and the number of files in each directory. The following image shows how the DelimitedPartitionReceiver parameters was set up for this use case.
After creating a new Scenario and downloading the scenario to a local machine, the GenRocket Advanced REST Client was run to pass the parameters to the Partition Engine and 10 GenRocket instances were launched.

The GenRocket Partition Engine, working with DelimitedPartitionReceiver automatically created a directory structure to store the generated data and each of the 10 partitions stored 1 million rows of data.

Below you can see a sample of the data generated that met the use case specifications. And 16 minutes of data generation were reduced to 3 minutes and 13 seconds for 10 million rows of data.

Note: The Partition Engine can be used to generate hundreds of millions to billions of rows of data in minutes.

Note: The Partition Engine can be used to generate hundreds of millions to billions of rows of data in minutes.

---

IMPACT

- By using various combinations of Linked Generators, the customer was able to design and model data sets for training and testing of an Artificial Intelligence / Machine Learning application.
- GenRocket generated 10 million rows of conditioned training data in a little over 3 minutes.
- This solution greatly reduced the time and effort of creating unique data sets that were useful for the customer use case at a much lower cost.

If you would like to know more about GenRocket’s Test Data Generation platform and our industry solutions, please visit our website at www.genrocket.com.