**Machine Learning .Net**

**Machine Learning**

* Systems that use algorithms and data to recognize patterns in data.
* ML technique that uses algorithms known as Neural Networks.
  + Trained to recognize patterns, relationships, and trends.
  + Once trained, neural networks can make predictions or decisions based on new, unseen data.
  + Performance will be improved when it is exposed to more data.
* **Deep Learning**
  + Is a subset of Machine Learning that focuses on using neural networks with many layers (deep neural networks).
  + Require large amount of data to generalize well.
  + Commonly used for:
* Image, speech, and emotion recognition
* Chatbots
* Driverless vehicles
* Tailored experiences

(Automated recommendations for products or other services and to perfect customer experiences based on purchase histories, past behavior, and other data.)

* Personal digital assistants

(Voice-activated personal digital assistants use deep learning to understand speech, respond appropriately to queries and commands in natural language)

* **2 Main Paradigm Machine Learning**
  1. **Supervised Learning**
* Algorithm is trained on a labeled dataset, meaning that the input data is paired with corresponding output labels or target values.
* The goal is for the algorithm to learn a mapping from inputs to outputs,
* Classification:
  + - * Predicting a categorical label or class for input data.
      * Data classification (like label the severity of customer reported issue.)
      * Image classification
      * Object classification
* Regression (Price prediction, Forecasting)
  1. **Unsupervised Learning**
* Algorithms are given input data without explicit output labels or target values.
* The algorithm's objective is to explore the inherent structure or distribution of the data.
* There is no "correct" output during training, and the algorithm must find patterns or groupings on its own.
  + Clustering
    - Grouping similar data points together based on their inherent similarities
    - K-Means (One of the most popular clustering algorithms)
  + Anomaly detection (Predicting items that don’t fit the rest of data)

**Use Cases for Each Task**

1. **Clustering**

Customer Segmentation

* Identify distinct groups of customers based on their purchasing behavior.

Image Segmentation

* Divide an image into meaningful segments or regions based on pixel similarities.

1. **Anomaly Detection**

Credit Card Transaction Fraud

* Find out this one purchase doesn’t match all other purchase that done in the past, then will going to flag this as a fraud.

1. **Regression**

Predicting House Prices

* Estimate the price of a house based on features such as square footage, number of bedrooms, and location.

Temperature Prediction

* Forecast the temperature for a specific location based on historical weather data.

Sales Forecasting

* Predict future sales based on historical sales data, seasonality, and marketing efforts.

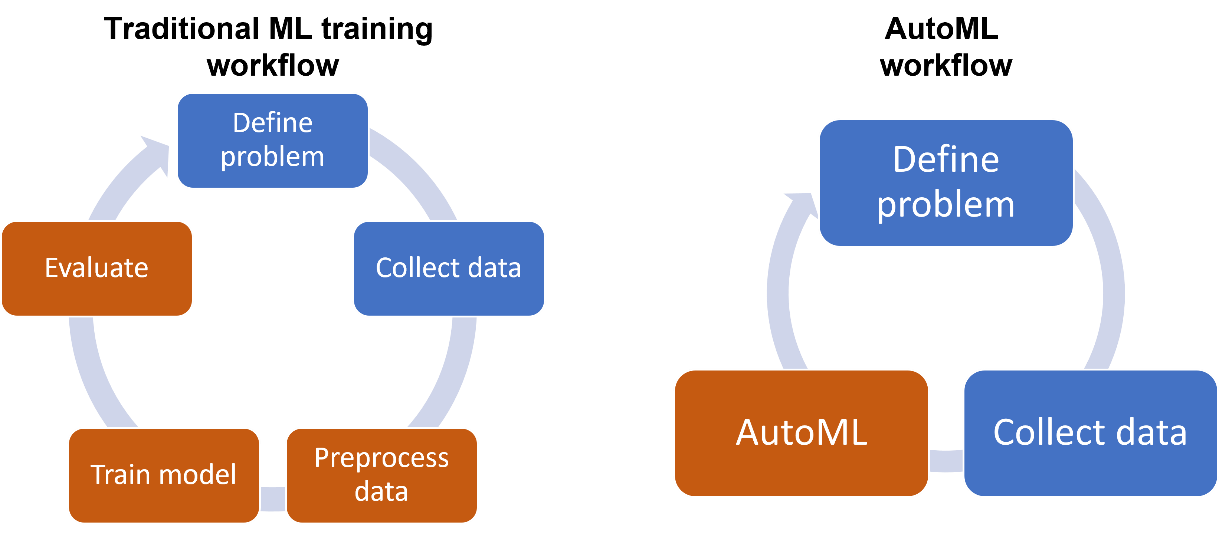
1. **Classification**

Credit Risk Assessment

* Classify individuals into categories such as "Low Risk" or "High Risk" based on their credit history and financial information.

**What is ML.Net?**

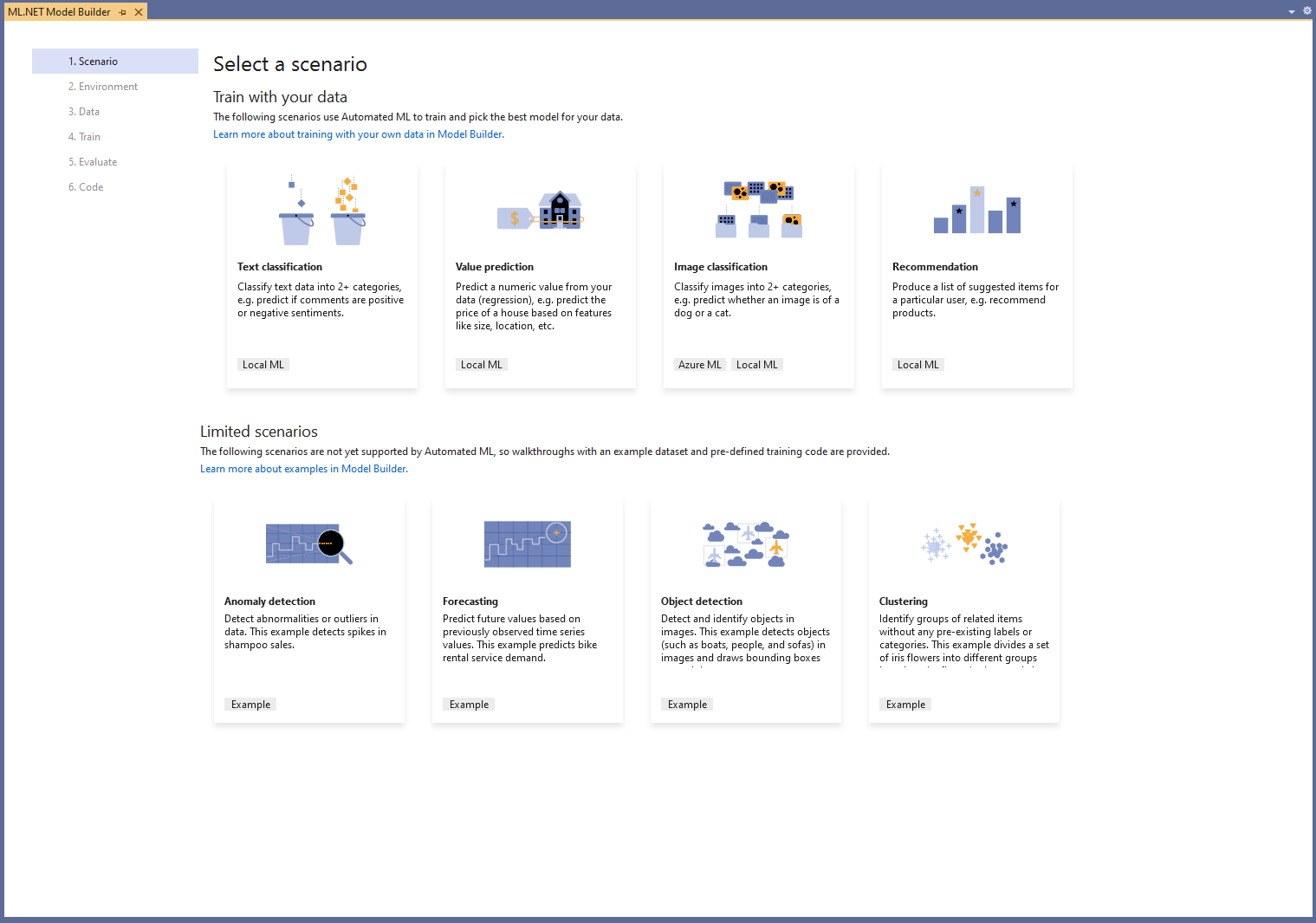
1. Open-source machine learning framework that enables developers to build custom machine learning models using .NET programming languages such as C# and F#.
2. Work well with large datasets.
3. Build models for a variety of use cases, including image classification, sentiment analysis, and fraud detection.
4. ML .Net includes features and tools that can integrate machine learning into your applications even more easily.
5. Features (Frameworks and libraries)
   * ML.NET
   * Automated Machine Learning (AutoML)

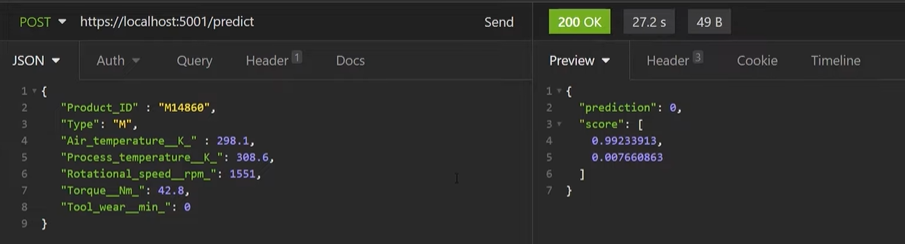


|  |  |  |
| --- | --- | --- |
| **Aspect** | **Without AutoML** | **With AutoML** |
| Model Selection | Manual selection of models based on expertise | Automated exploration of models to find the best one |
| Feature Engineering | Manual feature selection and transformation | Automated handling of feature engineering tasks |
| Data Preprocessing | Manual cleaning and preprocessing | Automated handling of data preprocessing tasks |
| Workflow | More manual intervention in the machine learning pipeline | Streamlined and automated workflow |
| Time and Resource Efficiency | May require more time and resources | Saves time and resources through automation |
| Accessibility | Requires expertise in data science and machine learning | More accessible to individuals with less expertise |
| Customization | Offers more control and customization | Provides a balance between automation and customization |

* + TorchSharp (TensorFlow.NET)
* TorchSharp is a .NET library that provides access to the library that powers PyTorch.
* Object Detection, Name Entity Recognition and Question Answering.
  + Open Neural Network Exchange (ONNX)
    - Common file format to enable AI developers to use models
  + Plotly.NET
    - Provides online graphing, analytics, and statistics tools.

1. Tools
   * Model Builder
     + Provides an easy-to-understand visual interface to build, train, and deploy custom machine learning models.



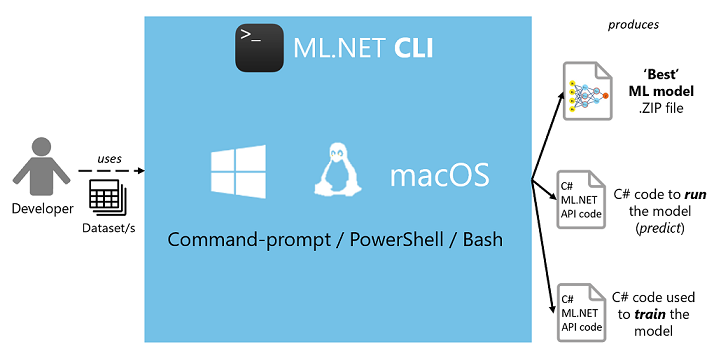
* + - Can work with csv files, tsv files, and SQL Server databases.
* Also, can make the model as API (Example of Device Failure Predictor) A screenshot of a computer

  Description automatically generated
  + - Model Builder produces a trained model, plus the code needs to load the model (.zip file) and start making predictions.

A screenshot of a computer program

Description automatically generated

* + ML.NET CLI (Command line interface)
    - Usually used with AutoML
    - It can generate those assets from your own datasets without coding by yourself.
    - It is simple to generate a high quality ML.NET model (.zip file) plus the sample C# code to run that model. In addition, the C# code to train that model is also generated, so that you can research and iterate on the algorithm and settings used for that generated "Best model".

<https://learn.microsoft.com/en-us/dotnet/machine-learning/automate-training-with-cli>

A screen shot of a computer

Description automatically generated

**Pros of ML .Net**

1. **Easy Integration and Compatibility**

* Supports multiple programming languages, including C#, F#, and Visual Basic.
* Integrates seamlessly with existing systems and libraries, allowing developers to use pre-existing code in their machine-learning applications without having to rewrite everything from scratch.
* Cross-platform development which can build applications on Windows, macOS, or Linux.
* Runs on different devices without issues.

1. **Performance**

* Designed to be lightweight and performant, making it suitable for applications with performance requirements, including machine learning tasks.

**Cons of ML.Net**

1. **Learning Curve for Some Languages**

* If developers are not familiar with C# or F#, there might be a learning curve in adopting these languages for machine learning tasks.

1. **Community Size Compared to Python**

* There might be fewer resources, tutorials, and community support specifically tailored to .NET for machine learning.

**What is Customer Segmentation?**

* Group customers according to various characteristics (for example grouping customers by age).
* It’s a way for organizations to understand their customers.
* It’s easier to make strategic decisions regarding product growth and marketing.
* Starting from the basic criteria, like gender, hobby, or age, it goes all the way to things like “time spent of website” or “time since user opened our app”.
* Bring advantages on budgeting, product design, promotion, marketing, and customer satisfaction.

**4 Types of Parameters**

1. **Geographic**

* All about the user’s location (group by country, state, city, or zip code)
* Preferred language

1. **Demographic**

* Many companies use gender differences to create and market products.
* Parental status is another important feature.
* Occupation.
* Can obtain data like this from customer surveys.

1. **Behavioral**

* Based on past observed behaviors of customers that can be used to predict future actions.

1. **Psychological**

* Deals with things like personality traits, attitudes, or beliefs.
* This data is obtained using customer surveys.

**Machine Learning for Customer Segmentation**

* Analyzing customer data and finding insights and patterns.
* **Improved accuracy**

Machine learning algorithms can analyze large datasets and identify patterns that may not be evident through manual analysis, leading to more accurate segmentation.

* **Ease of retraining**

Data keeps changing after the model is deployed. Usually, more labeled data becomes available after development and great resources will improve the performance of the model.

**There are 2 ways to update Customer Segmentation models:**

* + Use the old model as the starting point and retrain it.
  + Keep the existing model and combine its output with a new model.
* **Better scaling**

Machine learning models deployed in production support scalability which is flexible for future changes and feedback.

E,g. Considers a company that has 10000 customers today, and they’ve implemented a customer segmentation model. After a year, if the company has 1 million customers, then ideally, developers don’t need to create a separate project to handle this increased data. Machine Learning models have the inherent capability to handle more data and scale in production.

* Machine learning algorithm that’s suitable for customer segmentation problems is the k-means clustering algorithm.

**Steps to Implement Customer Segmentation with Machine Learning**

1. **Data Collection**

Gather data of customers such as demographics, transaction history, and preferences. This data can be collected from various sources like CRM systems, customer surveys etc.

1. **Data Preprocessing**

Clean and preprocess the data, removing any inconsistencies, missing values, or outliers that could negatively impact the accuracy of the segmentation.

1. **Feature Engineering**

Identify the most relevant features for segmentation, such as age, income, and purchase frequency. Consider creating new features that may capture additional insights. For example, you might create a feature representing the average transaction value or the recency of purchases.

1. **Model Selection**

Choosing a Clustering Algorithm, then selecting a suitable clustering algorithm for customer segmentation. Common algorithms include K-Means, Hierarchical Clustering.

1. **Model Training**

Train selected model on the preprocessed data, adjusting hyperparameters as needed to optimize the segmentation results.

1. **Model Evaluation**

Analyze the resulting customer segments to ensure they are meaningful and actionable. If the segments are not meaningful or actionable, then might need to revisit previous steps. This could involve adjusting features, trying different algorithms, or refining the preprocessing steps.

1. **Implementation**

Apply the customer segmentation to your marketing strategies, product development, and customer service efforts, tailoring your approach to better serve the needs of each segment.

**Machine Learning vs Deep Learning for Customer Segmentation**

**Clustering in Machine Learning:**

**Advantages**

* Provide clear groupings of similar customers, making it easier to interpret and understand the segments.
* Many clustering algorithms are relatively straightforward to implement and may not require as much computational power as deep learning.

**Considerations**

* Clustering methods assume linear boundaries between clusters, which may not capture complex non-linear relationships in the data.
* In some cases, might need to manually engineer features to represent customer behavior adequately.

**Tailored Experiences in Deep Learning:**

**Advantages**

* + Deep learning models can capture complex non-linear relationships in data, which can be beneficial for understanding intricate patterns in customer behavior.
  + Reducing the need for manual Feature Engineering.

**Considerations**

* Deep learning models often require large amounts of data to generalize well. If have limited data, overfitting may become a concern.
* Might need access to powerful hardware or cloud resources.

**References**

ML.NET CLI for Image Classification

<https://www.youtube.com/watch?v=B2wuYGnVKrk>

ML.NET Model Builder

<https://www.youtube.com/watch?v=cUqNzZwzUV0&t=141s>

Guide to install CLI

<https://learn.microsoft.com/en-us/dotnet/machine-learning/how-to-guides/install-ml-net-cli?tabs=windows>