Intelligently prioritize and route tasks across department with Agent Intelligence.
# Table of Contents

Outline and Overview .................................................. 3

The need for intelligent automation ................................. 3

Fulfiller and Requestor Experience .................................. 5

What is Agent Intelligence ............................................. 6

How are solutions trained? ............................................. 6

How do predictions happen? ........................................... 8

Agent Intelligence – Best Practices .................................. 9

1. Pre-requisites to using Agent Intelligence .................. 9
   1.1. Identify the right problems you want to solve ........ 9
   1.2. Do you have enough data? ................................. 9
   1.3. Do you have the right data? ............................... 9
   1.4. Have right expectations .................................. 10

2. Best Practices to implementation process .................. 10

3. Best Practices to Configuring Agent Intelligence Solutions .................................................. 11
   3.1 Choose the right filters for your data set ................ 11
   3.2. Choose the right input fields ............................. 12
   3.3. Try multiple solution configurations ................... 12

4. Best Practices to Using Agent Intelligence Solutions .......... 12
   4.1. Identifying a good solution ............................... 12
   4.2. Tuning a trained solution ................................. 13
   4.3. Using trained solutions for Prediction ................ 13
   4.4. Ongoing solution management.......................... 13

Conclusion ................................................................. 14

Glossary ................................................................. 14
Supported Versions

This whitepaper ©2018, supports Agent Intelligence for release in Kingston and London.

Disclaimer

ServiceNow is transforming the way people work. ServiceNow is ushering a breakthrough approach to machine learning by making it more accessible to organizations—without relying on an army of data scientists. We have built machine learning capabilities directly into the Now Platform™. Machine learning promises to deliver new levels of automation—Empowering people to get work done smarter, and faster. Now, you can use data that you already have to create tuned models that intelligently prioritize, and route tasks across departments. The content covered in this whitepaper represent Servicenow integration and assumptions as of today. We undertake no obligation, and do not intend to update the forward-looking changes, to review, or confirm expectations on installations completed. Further information on these, and other factors that could affect our technical outcomes are not intended to be included.

Outline and overview

ServiceNow has introduced a breakthrough approach to making machine learning more accessible to organizations – one that operationalizes machine learning solutions within your existing processes without the need for an army of data scientists to build custom solutions. The machine learning capabilities are powered directly by the Now Platform™.

This document presents an overview of Agent Intelligence(Al) specific to problems it solves. We will also cover recommended best practices while implementing Agent Intelligence.

Agent Intelligence improves the interactions between processes and service agents. The goal of this document is to show how Agent Intelligence reduces IT incident triage time and customer service response times.

All the information presented here is based on automated intelligent systems leveraging machine learning processing. Using trained machine learning models, we will cover how automated categorization, and prioritization of the data can bring automation to ServiceDesk. This whitepaper will specifically cater to Process Owners, and Technical Architects.
The need for intelligent automation

Improving user experience
Manually triaging thousands of incidents, categorizing tasks, and routing work can consume peoples’ resources. The Accenture study that was recently completed illustrates the potential scale of this problem: 43% of IT service desk respondents, and 27% of customer service desk respondent have approximately 100+ different assignment groups to choose from. It can also introduce human error, and slow resolution time. Moreover, attrition and turnover of human agents causes loss of institutional knowledge, and it takes a long time for newer agents to become productive. Imagine an agent trying to figure out one of the hundred categories appropriate to the problem described by a 180-word description!

The main objective of Agent Intelligence is to reduce IT incident triage time, and customer service response times. ServiceNow's approach enables machine learning to decide on the category based on prior history learned from historical data.
Fulfiller and requestor experience

The challenge for front-line customer support staff is that they are unable to see meaningful patterns within these huge volumes of data. It is often difficult for the support staff to determine the right category, priority, and assignment group for a specific ticket—even if it’s a repeatable task like a server reboot. The individual customer context is vital. Understanding this context often takes time. It is complicated by the fact that every individual will approach the question in their own way, based on their particular skills, experience, institutional knowledge, and time pressures. This is where machine learning is extremely powerful. When machine learning is applied to the categorizing and assigning of tickets—a service organization can derive multiple productivity, plus quality benefits.

Agent Intelligence facilitates organizations to use machine learning algorithms to set field values during record creation, such as setting the incident category based on the short description. You can also train predictive models to act as an agent, for automatically categorizing information, and routing work based on your past record-handling experience. It is learned behavior applied without bias.

ServiceNow Agent Intelligence is available on our Kingston and London releases on both IT Service Management and Customer Service Management products and for HR Service Management in London.

Agent Intelligence for IT Service Management improves response times, lowers error rates and reduces costs. Agent Intelligence delivers machine learning capabilities without the need for data scientists by allowing automatic incident categorization and assignment. It allows organizations to monitor prediction based on the unique characteristics of each customer’s data.

Agent Intelligence for Customer Service Management uses machine learning to accelerate case categorization, prioritization, and assignment. Thus, shortening response times, and contributing to higher customer satisfaction. It also reduces manual work, freeing up people resources to work on higher priority tasks.

Agent Intelligence for HR Service Management uses machine learning to pick the right HR Service and COE for an email to create the correct Case thus shortening the time it takes to resolve the case.
What is agent intelligence

Agent Intelligence is embedded and available within the ServiceNow platform and uses supervised machine learning to automate prediction of fields like category and assignment group for incidents and cases. Rather than manual triaging of tasks like IT incidents and Customer Service Cases or relying on assignment rules which have to be created and managed manually – Machine Learning automates this process by learning from data. Agent Intelligence allows configuration and easy deployment of machine learning solutions by business users rather than requiring specialized data scientists to operationalize this.

Supervised machine learning requires that sufficient historic data exist with examples of different classes (categories, assignment groups etc. that have to be predicted), so that they can be used for training the solution. For example, in the case of incident categorization, there should be enough number of closed incidents which have the right categories assigned to these incidents for a machine learning solution to be built.

How are solutions trained?

Solutions need to be configured before they can be trained. A solution definition is a configuration that specifies the input data, and prediction outcome for training a solution.

Our customers are granted the right to use Agent Intelligence solution definitions provided by ServiceNow, and to create new solution definitions for any ServiceNow table. Pre-defined solution templates are provided out-of-the-box for predicting incident category and assignment group fields for IT Service Management, predicting case category, assignment group and priority fields for Customer Service Management and predicting HR Service for HR Service Management. Customers can extend Agent Intelligence to other processes by creating their own solutions and training them on their historical data. We will look at the best practices for configuring solution definitions in section 3.

Once a solution is configured, you can train and activate the solution so that it can be used to make predictions when new incidents or cases are created.

The training process involves extraction of raw data that is specified in the solution definition and sending it to a shared training server in ServiceNow data center where the solution will be trained.

Agent Intelligence allows configuration and easy deployment of machine learning solutions by business users rather than requiring specialized data scientists to operationalize this.
As part of the training process, the input data is processed in the following ways to generate the analytical dataset, thereby preparing the data to be run through statistical or machine learning models:

• Data is cleaned removing any problematic characters;
• Duplicate records are identified and indexed
• Text input, like short description is tokenized and broken into n-grams (simply put, n-grams are all possible combinations of adjacent words).
• Stop words in text input are removed since these words do not generally influence how the record should be classified
• Stemming, a process which identifies the root words for the given text input is performed

In London release, we support language specific processing for steps like stop word removal and stemming to be applied for datasets that are predominantly in French, German, Spanish, Dutch or Japanese languages by allowing the users to specify the processing language in the solution definition (English processing is applied by default to all data-sets).

The analytical dataset is then input into classification algorithm which identifies patterns of data that are associated with different classes (for example, categories or assignment groups) and generates the underlying predictive models for a solution. The solution is then shipped back into the customer instance. In the process, any underlying customer data is removed from the training server. Once the solution is available on a customer instance, it can start making predictions.
How do predictions happen?

Agent Intelligence provides a prediction API that is invoked on a trained solution to make a prediction. By default, Agent Intelligence is invoked through business rule “Default Incident Based Prediction” or “Default Case Based Prediction” that is provided out-of-the-box. So, when a user types the short description on an incident form and submits it, the default business rule is triggered which calls the prediction API. The input data, of the incident is passed to the solution and the underlying model identifies the most likely class for this incident or case or other task that you have built the solution for. The predicted value is saved on the output field of the record. Note that customers can configure the business rule to suit their implementation or even call the prediction API from a different script/workflow.
Agent intelligence – best practices

This section of the whitepaper highlights the best practices to configure and use Agent Intelligence to achieve maximum benefits.

Before you start your implementation journey, it is important to understand whether you are prepared to make use of Agent Intelligence, and the power of machine learning by understanding what the pre-requisites are.

1. Pre-requisites to using agent intelligence

Agent Intelligence makes use of your historic data to learn patterns about your business processes. For example, how do you categorize your incidents, so that it can predict the category in the future. There are few pre-requisites to start using Agent Intelligence, and reaping the benefits it offers.

1.1. Identify the right problems you want to solve

Before you build solutions using Agent Intelligence, it is important to identify what problems you want to solve with AI. We have a few solutions that are designed out of the box. For example, for incident management, we have solution definitions for Incident Categorization and Incident Assignment group predictions. While you could make use of these solutions, you could build custom solutions to predict other categorical/choice fields if you have the data to train such solutions. For example, would it be helpful to predict the urgency of an incident rather than ask the user to provide it. Would it be helpful to predict the Configuration Item class or Business Service of an incident? Identify the right problems so that you can build the right solutions. As long as the field you want to predict is a choice field with finite number of choices (2 to few hundreds), they would be a candidate for automatic prediction using Agent Intelligence.

1.2. Do you have enough data?

Although how much data is required depends on the specific problem you are trying to solve, we recommend that you have at least 30,000 to 300,000 historic records from which AI can learn. So, for building a solution that can predict the category field of an incident record, you need these many closed incidents which have the category field populated.

1.3. Do you have the right data?

While having sufficient quantity of data is important, what is more important is that the data is accurate. For example, if your closed incident records do not have the right category assigned to them, AI will be learning off those records, and the predictions it would make will be wrong as it would be based on wrong learning. Making sure that you have a process to ensure that fields are correctly marked on the records will help in ensuring that when you decide to use AI, and machine learning, you can get the right results.
1.4. Have right expectations

It is important to have right expectations when you implement AI. AI will not be able to predict for 100% of the incidents with 100% accuracy. As long as the solutions you build can match human accuracy, they would be of significant value because you would be automating repetitive work that is being manually done with the additional benefit of not having to retrain new employees to pick up these skills every time there is an employee churn. It is important that you have a well-defined process to address scenarios where AI may not make a prediction. Based on tests across multiple customer datasets, a target of 70% for both precision and coverage is a good starting point to your AI journey.

2. Best practices to implementation process

Once you have established the pre-requisites outlined in the previous section, following a well-designed process is important to having a successful implementation. The overall end to end implementation process should not take more than 2 weeks if correctly executed. We recommend that you do all necessary testing on a sub-production environment before you deploy in Production – however please note that it is extremely important to make sure that on Day 1 of your implementation journey, you begin with cloning your production data on to your sub-production environment.

From day 2 to day 10, we recommend testing various solutions for your use cases. Use the best practices to configure and use solutions in the next 2 sections to identify the right solution that you want to use in production.

On Day 11 and Day 12, test the final solution(s) you have decided to implement in production. Note that if you are on London release, you can use an update set to easily export solution you have already trained in sub-production environment to your production, this should make this process even faster.

From day 14 onwards, follow the best practices outlined in ongoing solution management section below to get best results.
3. Best practices to configuring agent intelligence solutions

3.1. Choose the right filters for your data set

“Garbage in, Garbage Out” is the old saying, and rings very true when you begin your journey with Machine Learning. The end goal should be to maximize the accuracy of the model against the greatest number of records, which is only possible if the data used for training the solution is based on useful data. Too many choices, inaccurate selections, or old process results can reduce the solution’s accuracy, or limit the numbers of records that it’s applicable against.

1. Ensure the training data reflects only the accurate business processes of today’s environment. More data doesn’t always mean a better model. Identifying historic data that is representative of the data you expect during predictions is a great first step in this process. If there was a major change in your process 6 months back, it is advisable to use data from this period. Here are a few examples of major changes:

   • You launched a new portal for end users introducing a new channel for incidents.
   • You redesigned the incident form significantly by changing what fields are mandatory and what fields are optional for users to enter.
   • You restructured your service desk by introducing a completely new org in a specific region.
   • Your company merged with another major player expanding your portfolio significantly impacting the kind of incidents that can come in.

2. Remove inactive categories, or assignment groups in your historic dataset using a filter. For example, if the assignment group of “Service Desk, Ireland” is no longer valid, add a filter “Assignment Group is not Service Desk, Ireland” to remove such historic records from being used for training the solution. This ensures that the model is not training on stale data, and it will not predict these classes when incidents come in.

3. Another consideration is to evaluate what incidents/cases would be useful from a prediction perspective, so that only such type of records are used when training the solution as well. For example, if your organization has a large pool of machine generated incidents which you don’t find useful to predict categories/assignment groups for such incidents, you can remove these incidents from training the solution.
3.2. Choose the right input fields

The out-of-the-box solution definitions use short description of an incident as an input field for the category/assignment group solutions. If you have another mandatory field that a user always provides when creating a new incident, you could potentially use it as an additional input field. Consider building two solutions one with just the short description as input, and another with short description, plus the additional input field. You could compare the estimated performance of the solutions to identify the best input field combination for your solution.

3.3. Try multiple solution configurations

A key consideration when configuring a solution definition is to choose how far back we go in time for choosing the dataset. It is a good practice to try more than one solution by just varying the time filter. For example, different solutions can be trained by changing the filter to use incidents closed/resolved in last 3 months versus last 6 months, or 9 months. The key consideration for the filter is to make sure there are at least 30,000 records. You could also try other solution configurations by changing the input fields for the solution.

A recommended practice is to create a new solution definition whenever you are trying a new combination of input/output fields but use the same solution definition to modify the filters to generate different solutions so that you don’t proliferate with too many solution definitions. Once the different solutions are trained you could identify which solution has the better estimated performance as explained in 4.1.

4. Best practices to using agent intelligence solutions

4.1. Identifying a good solution

As part of solution training, two key metrics – Estimated Solution Precision, and Estimated Solution Coverage are generated for the solution. Coverage indicates what percentage of the records the solution can predict, and Precision indicates what percentage of the predictions will be correct. By comparing these metrics between different solutions users arrive at the right solution to put to work. We want to achieve a balance between the number of incidents we guess against, and how accurate we are. Think goldilocks. If the solution is set to only predict when it is very sure (high precision) then it will make less predictions. If the solution is set to always make a prediction (high coverage) then it will be less accurate when it does.
4.2. Tuning a trained solution

Once the solutions are trained, and you have identified the best solution based on overall estimated performance in Stage 3, it is time to examine the solution to check if it needs to be tuned for improving the performance. In some cases, it might mean you want to improve precision. In some other cases this might mean that you want to increase what proportion of the incidents the solution is able to predict.

Identify a class where precision is poor (which means that the accuracy of predictions for this class is low). You can look at the possible precision-coverage combinations for this class and apply a combination where precision is higher, but coverage is lower. By doing this, when a new incident comes in that belongs to this class, the solution will make a prediction when it has higher confidence than before thereby increasing the precision.

Identify a class where the coverage is poor (which means that the solution will skip predicting in more cases). There could be classes where you don’t mind a lower precision as long as the solution can predict for a greater number of incidents. Again, you can look at the possible precision-coverage combinations, but this time—pick a combination that has a higher coverage but lower precision.

4.3. Using trained solutions for prediction

Once you have identified the right solutions you want to use for prediction, you will have to ensure that the out-of-the-box business rule can be used as is, for making predictions before turning it on in production.

Firstly, make sure only the solutions you want to use for prediction are active – any other solutions you might have created for testing purposes must be deactivated. Secondly, make sure that you have a process for triaging tasks for which a prediction cannot be made – i.e. if your solution coverage is 70%, make sure that you have a process for the remaining 30% incidents (or cases) for which a prediction is not made. For example, if you were relying on a service desk to manually triage all records, they could triage these 30% tickets going forward. Or if you had assignment rules to route tasks, make sure that the order of your prediction business rule is correctly set, so that the assignment rules can apply for the remaining 30% of records only.

4.4 Ongoing solution management

As your business processes change, it is important to make sure that you retrain your AI solutions to adapt to these facets of your business. You could retrain a solution when such changes occur, or you could setup retraining based on a frequency – for example, every 30 days or 60 days. Retraining a model is useful only when there is sufficient new data that you could use for learning purposes. A recommended practice is to retrain a solution atleast once a quarter or when there is a minimum of 10% new data in your training set.

Another way to determine when a retraining might be required is to look at your Prediction Results report which monitors the ongoing coverage, and precision for every solution that has been deployed. When you notice a drift, or a consistent drop in the coverage, or precision of a solution—you know that it is time to retrain the solution.

Also, when you have changes in your process—for example—you have introduced new categories into your process, and there have been sufficient number of incidents that have been added into the system which belong to these categories, it is a good idea to retrain the solution so that it can predict when a new incident comes which belongs to this category. Similarly, if you retire categories, or assignment groups adding a filter on your solution definition to remove these categories, and training the solution is a recommended practice so that these categories are not predicted.
Conclusion

As a part of ServiceNow’s intelligence automation strategy, we continue to address some of the top customer asks:

• “Help us improve our customer support centers.”

• “Help us improve our automation.”

• “Help us improve IT support.”

ServiceNow has also started to stretch the boundaries of targeted knowledge by introducing machine learning and artificial intelligence into the platform capabilities. We help organizations to improve speed, and accuracy while lowering costs.

Glossary

• AI – Agent Intelligence

• ITSM – IT Service Management

• CSM – Customer Service Management

• App – Application specific software product installed on (specific) infrastructure used to perform a function.

• Solution Definition – A configuration of an ML solution which defines what is the outcome/output field for a solution, what data is used for training this solution and what is the input field(s) used for training.

• Solution – A trained solution which contains the underlying predictive models, that is generated when a solution definition goes through the training process

• Precision – Percentage of predictions that are correct for an ML solution. An estimate of this metric is available when a solution is trained and it is referred to as “estimated precision”.

• Coverage – Percentage of incoming records that are predicted using an ML solution. An estimate of this metric is available when a solution is trained and it is referred to as “estimated coverage”.