

Mastering Model Optimization with Artificial Intelligence (AI)

WELCOME TO OUR WEBINAR



Mikhail Golovnya
*Senior Advisory Data
Scientist*



David Peralta
Area Marketing Manager

WEB-AUDIO:

Please make sure you have your computer audio system activated and your speakers turned up.

QUESTIONS:

You can enter your questions at any time in the questions section.

About Our Speakers:

Mikhail Golovnya

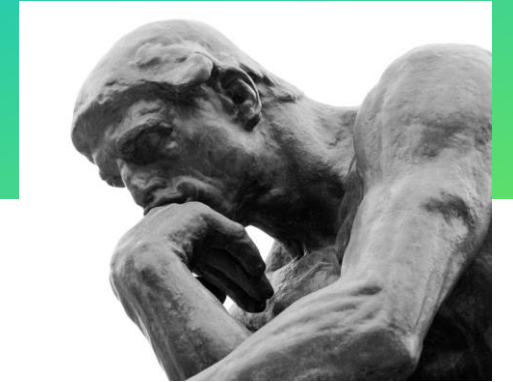
Senior Advisory Data Scientist

Mikhail is a Senior Advisory Data Scientist at Minitab. He has been prototyping new machine learning algorithms and modeling automation for the past twenty years.

Mikhail has been a major contributor to Minitab's on-going search for technological improvements among the most important algorithms in Machine Learning.



Things to Contemplate



87% of machine learning models never make it into production

-Venture Beat

Why?

85% of machine learning projects fail to deliver on their intended promises

-Gartner

Predictive Analytics Workflow



Deployment Approaches

How can we operationalize a predictive analytics solution (PA model) to a problem?

Traditional answer: encode model equation

Modern Answer: deploy model universally in Model Ops

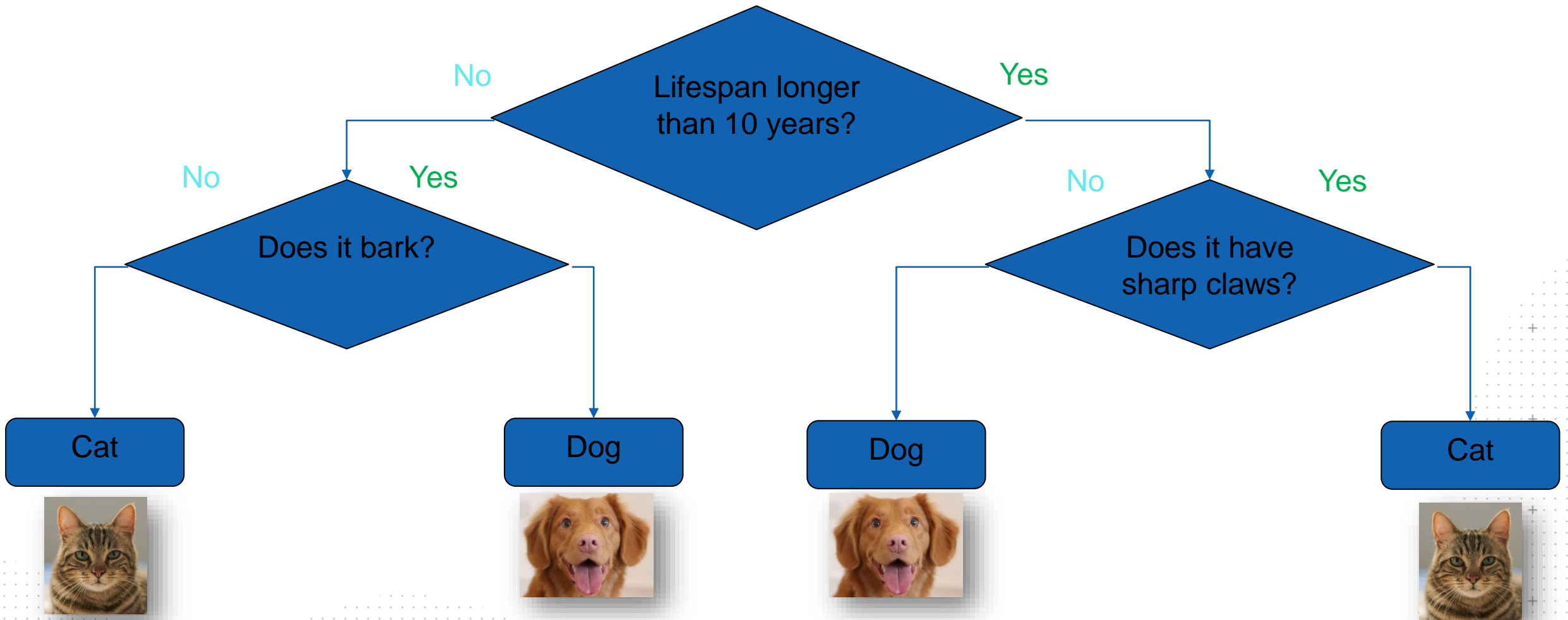


Survey 1

- ▶ How often do you need to operationalize your models?
 - Never
 - Sometimes
 - Always

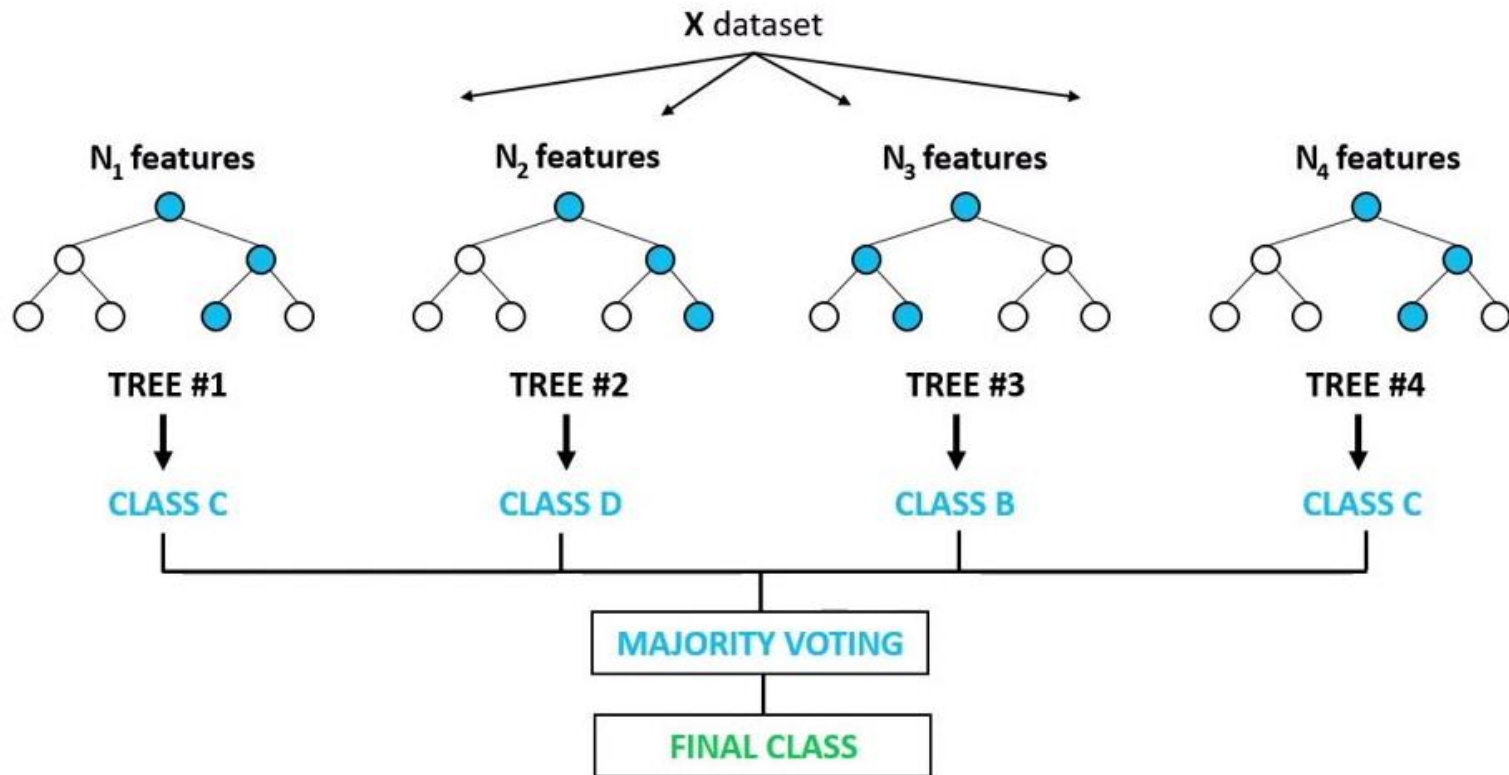
Equation-Based Deployment

CART Has No Equation

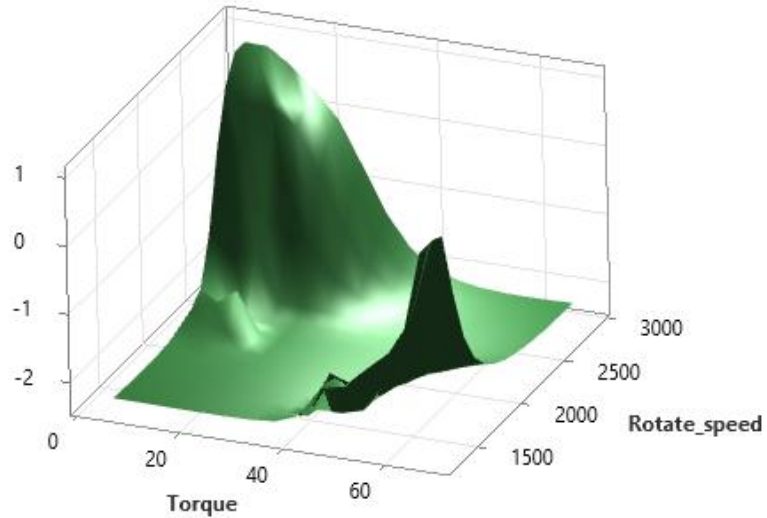
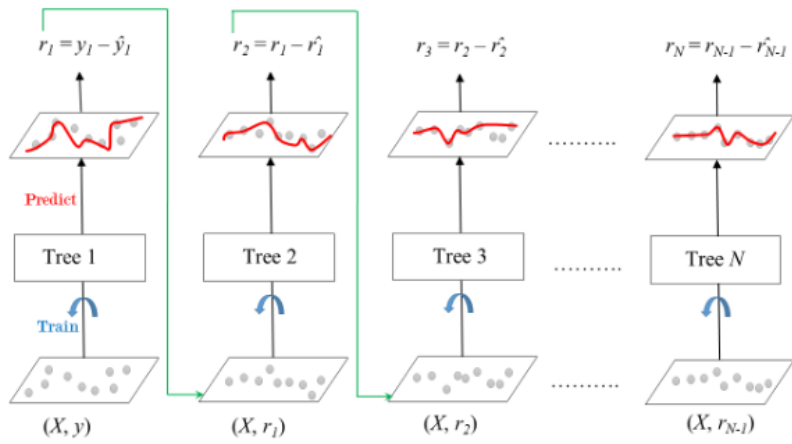


Random Forest is a Black Box

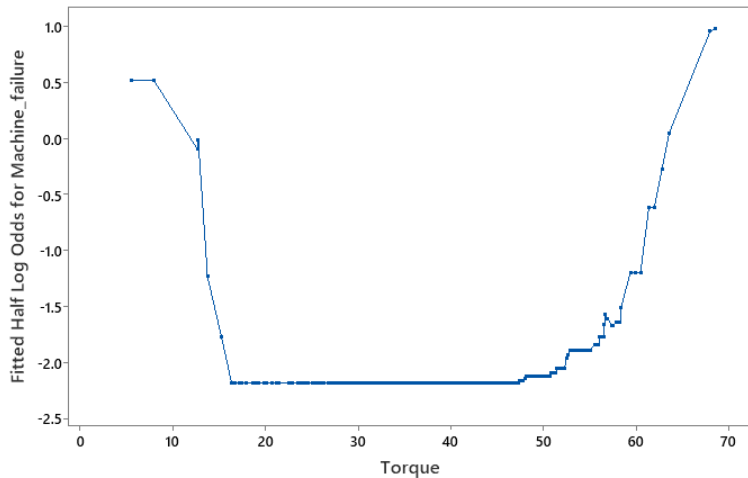
Random Forest Classifier



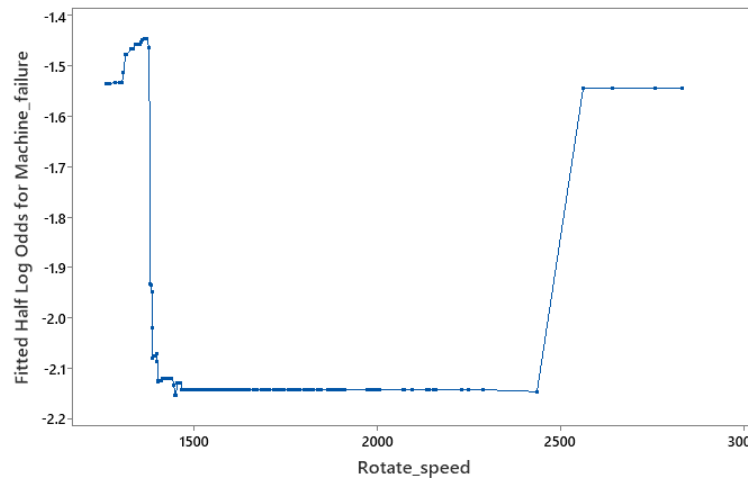
TreeNet is an Enigma



One Predictor Partial Dependence Plot



One Predictor Partial Dependence Plot

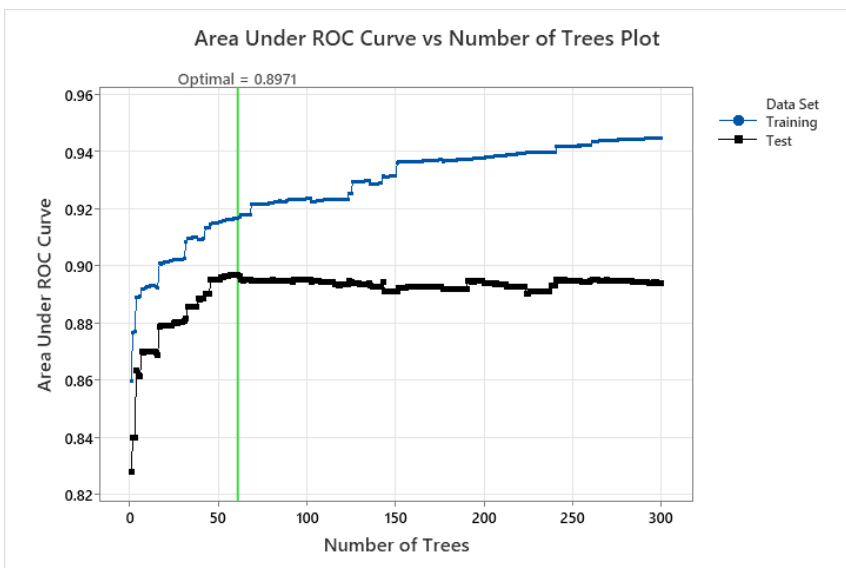


An Idea!

- Take the TreeNet plots and fit them with conventional parametric functions
 - Alternatively: construct splines using the cutoff points identified on the TreeNet plots
- Apply these functions to the original predictors as univariate transformations to obtain derived features
- Use the derived features to fit conventional linear or logistic regression
- Don't worry about getting the scaling right, let the regression do it



TreeNet Model (Classification)

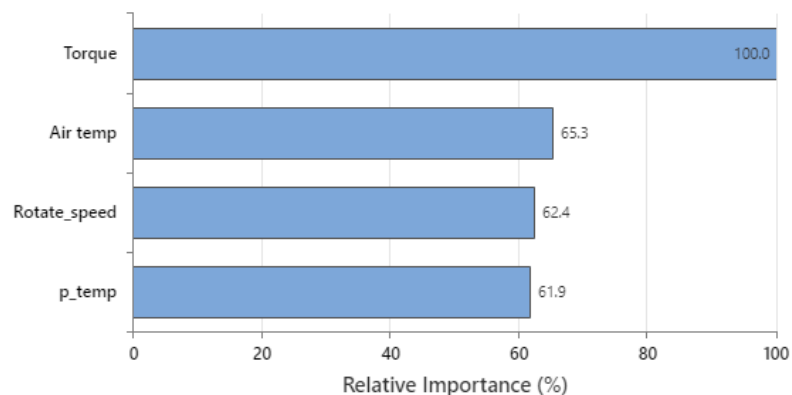


Model Summary

Total predictors 4
 Important predictors 4
 Number of trees grown 300
 Optimal number of trees 61

Statistics	Training	Test
Average -loglikelihood	0.0734	0.0798
Area under ROC curve	0.9168	0.8971
95% CI	(0.8924, 0.9412)	(0.8568, 0.9374)
Lift	8.0579	7.5258
Misclassification rate	0.0196	0.0222

Relative Variable Importance

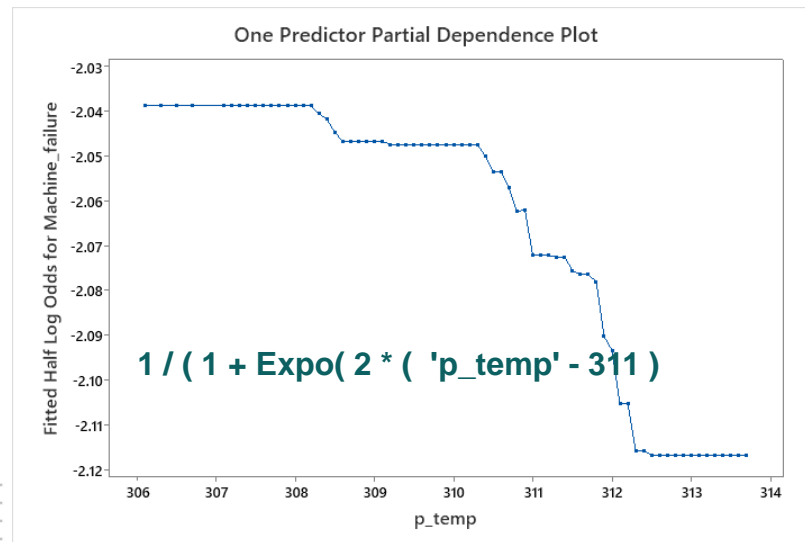
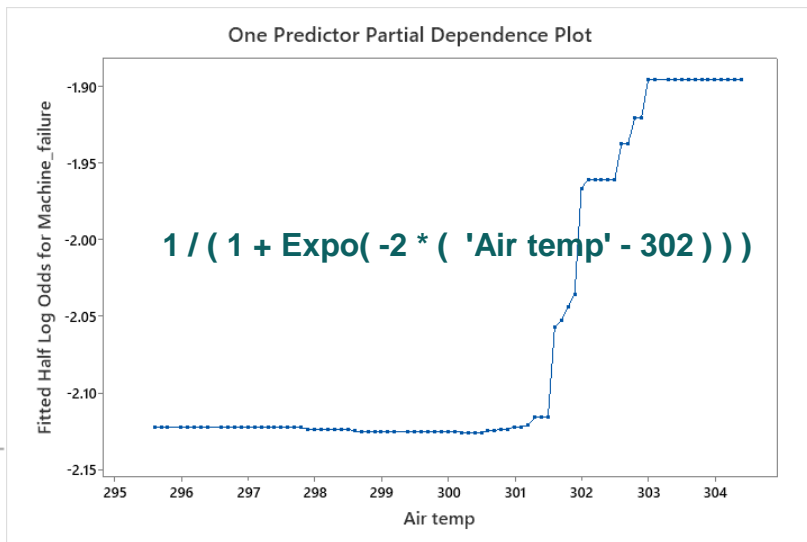
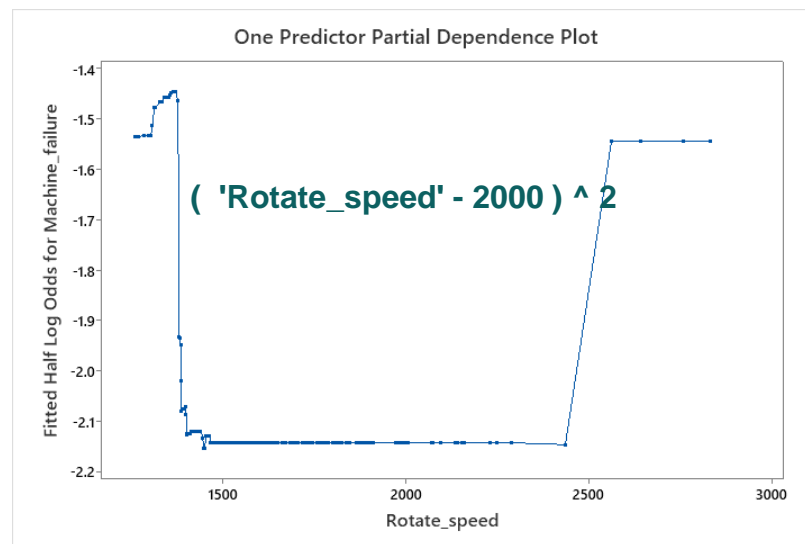
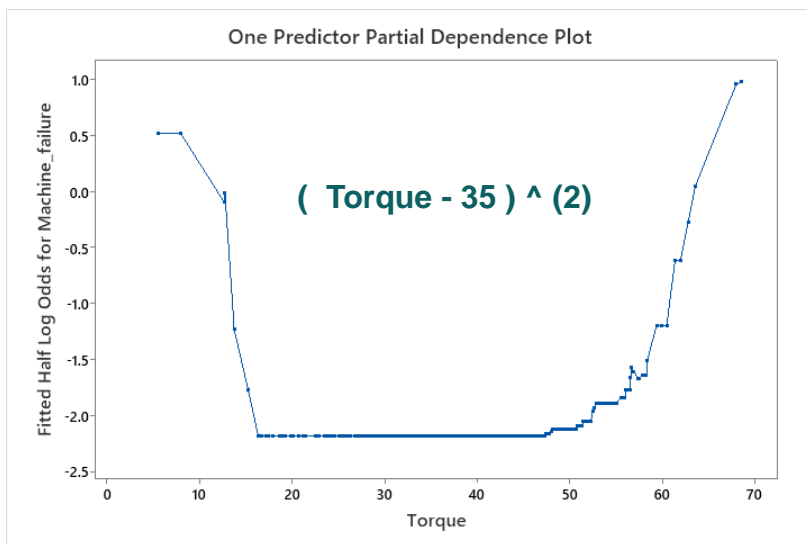


Compare Algorithms (ROC)

CART: 0.84
 RandomForest: 0.89
 TreeNet: 0.90



Guess the Equation!



Build a Conventional Model

- Original Model

Regression Equation

$$P(1) = \frac{\exp(Y')}{1 + \exp(Y')}$$

$$Y' = -32.6 + 0.6853 \text{ Air temp} - 0.660 \text{ p_temp} + 0.010728 \text{ Rotate_speed} + 0.2626 \text{ Torque}$$

Model Summary

Deviance R-Sq	Deviance R-Sq(adj)	AIC	AICc	BIC	Area Under ROC Curve	Test Deviance R-Sq	Test Area Under ROC Curve
30.58%	30.38%	1386.45	1386.45	1420.71	0.8606	27.08%	0.8571

- New Model

Regression Equation

$$P(1) = \frac{\exp(Y')}{1 + \exp(Y')}$$

$$Y' = -8.307 + 3.547 \text{ AirTempT} + 2.270 \text{ p_tempT} + 0.000003 \text{ Rotate_speedT} + 0.005330 \text{ TorqueT}$$

Model Summary

Deviance R-Sq	Deviance R-Sq(adj)	AIC	AICc	BIC	Area Under ROC Curve	Test Deviance R-Sq	Test Area Under ROC Curve
34.93%	34.72%	1300.28	1300.29	1334.55	0.8897	33.16%	0.8743

Summary of the Procedure

- We can use **TreeNet** to **automatically** discover the nature of non-linearities and then use this knowledge to improve conventional models
- **Residual loss** can be assessed by comparing the original TreeNet model performance with the improved conventional model performance
- Not capturing **interactions** is the usual culprit to explain the residual loss in accuracy
- Another way to get the formula is to use MARS



Survey 2

- ▶ Do you need to deploy/operationalize your models outside of MSS (universal deployment)?
 - Yes
 - No

Universal Deployment

Minitab Model Ops – a comprehensive platform to deploy, manage, and monitor predictive analytics models

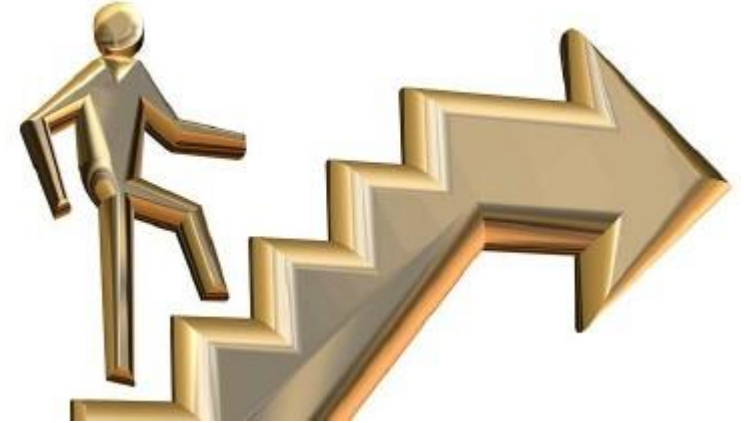
Bridge the gap between model creation and model production with a simple, yet powerful Model Ops platform.

Build your models in Minitab Statistical Software and deploy them in **Minitab Model Ops**.



Key Steps to Model Deployment

- **Import** (upload) a model into Model Ops
- **Deploy** (turn on) a model in Model Ops
- **Activate** (access) a model in Model Ops
- **Score** records in real time



Step 1: Importing a Model into Model Ops

The screenshot shows the Minitab Model Ops web interface. A modal dialog titled "Import Model" is open, allowing a user to upload a model file. The dialog contains the following fields and options:

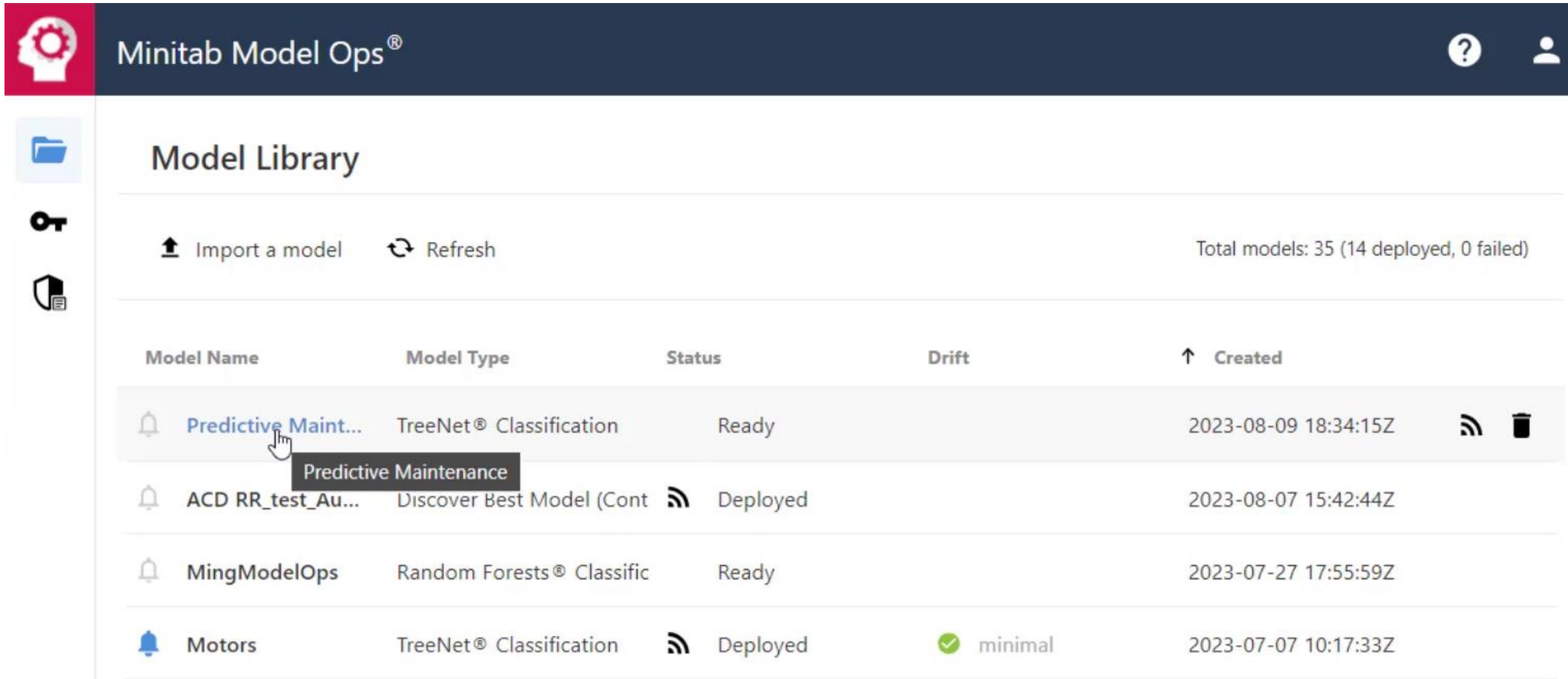
- Select a File:** A text input field containing "webinar.mpx".
- Model Name:** A text input field containing "Predictive Maintenance".
- Specify the model within the file:** A dropdown menu showing "TreeNet® Classification: Machine_fa...".
- Model Type:** A text label showing "TreeNet® Classification".

At the bottom of the dialog are two buttons: "Close" and "Import". A mouse cursor is hovering over the "Import" button.








In the background, the "Model Library" is visible, showing a list of models with columns for Model Name, Model Type, Status, and Last Updated. The list includes models like "ACD RR_test", "MingModel", "Motors", "Sami_Conso...", "MingMaintenan...", "Sami_Knock_Test2", and "Sami_Knock_Test".

Model Name	Model Type	Status	Last Updated
ACD RR_test			15:42:44Z
MingModel			17:55:59Z
Motors			10:17:33Z
Sami_Conso...			13:33:06Z
MingMaintenan...	CART® Classification	Deployed	2023-07-05 00:27:49Z
Sami_Knock_Test2	Multiple Regression	Deployed	2023-06-26 16:44:15Z
Sami_Knock_Test	Multiple Regression	Deployed	2023-05-29 15:06:21Z

The Model is Available





The screenshot shows the Minitab Model Ops interface. At the top, there is a header with the Minitab logo and the text 'Minitab Model Ops'. Below the header, there is a 'Model Library' section. On the left side, there are navigation icons for a folder, a key, and a document. In the main area, there are buttons for 'Import a model' and 'Refresh'. To the right, it says 'Total models: 35 (14 deployed, 0 failed)'. Below this is a table with columns: Model Name, Model Type, Status, Drift, and Created. The first row is highlighted, and a tooltip 'Predictive Maintenance' is shown over the model name 'Predictive Maint...'. The second row is 'ACD RR_test_Au...' with status 'Deployed'. The third row is 'MingModelOps' with status 'Ready'. The fourth row is 'Motors' with status 'Deployed' and a 'minimal' drift indicator.

Model Name	Model Type	Status	Drift	Created
 Predictive Maint...	TreeNet® Classification	Ready		2023-08-09 18:34:15Z
 ACD RR_test_Au...	Discover Best Model (Cont	 Deployed		2023-08-07 15:42:44Z
 MingModelOps	Random Forests® Classific	Ready		2023-07-27 17:55:59Z
 Motors	TreeNet® Classification	 Deployed	 minimal	2023-07-07 10:17:33Z



▶ The model is now available for deployment










Step 2: Deploying a Model

 Minitab Model Ops® ? 👤

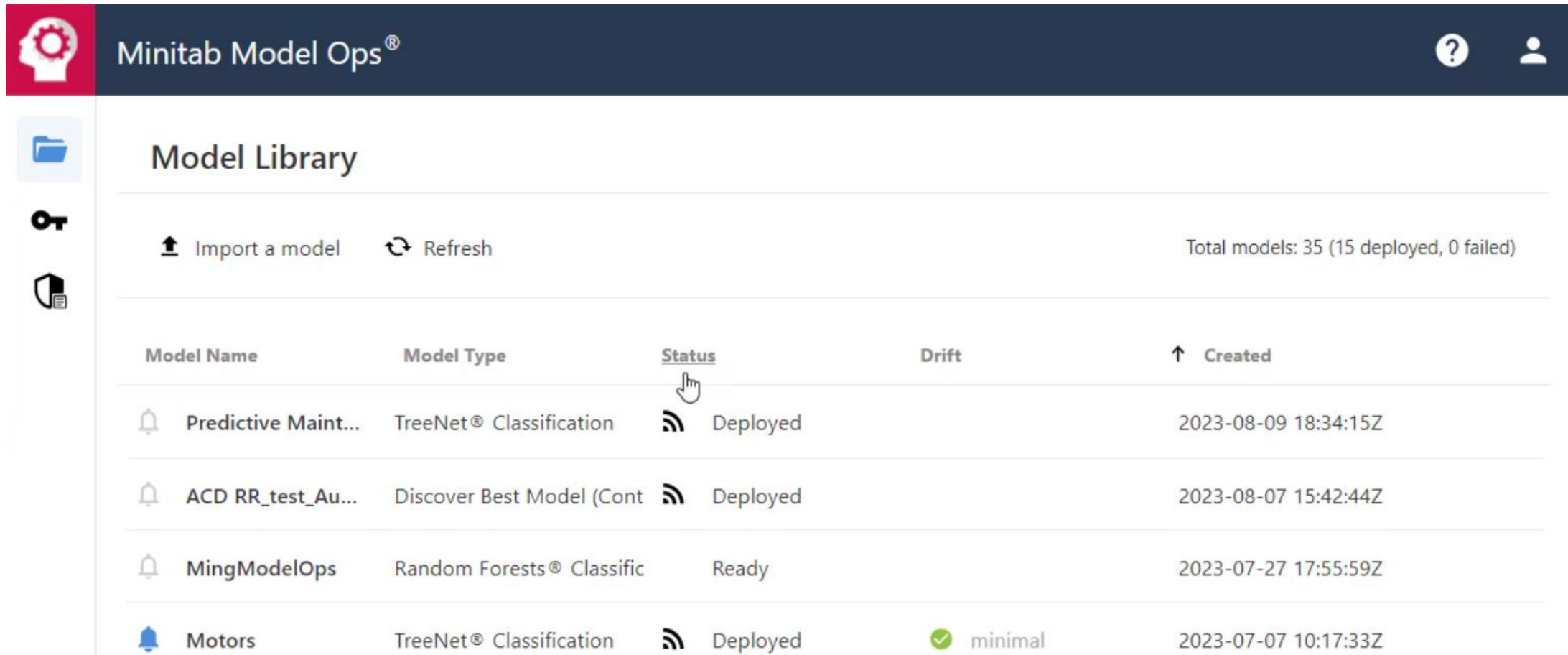


Model Library









 Import a model  Refresh Total models: 35 (14 deployed, 0 failed)

Model Name	Model Type	Status	Drift	Created	
 Predictive Maint...	TreeNet® Classification	Ready		2023-08-09 18:34:15Z	  Deploy
 ACD RR_test_Au...	Discover Best Model (Cont	 Deployed		2023-08-07 15:42:44Z	
 MingModelOps	Random Forests® Classific	Ready		2023-07-27 17:55:59Z	
 Motors	TreeNet® Classification	 Deployed	 minimal	2023-07-07 10:17:33Z	

The Model is Deployed

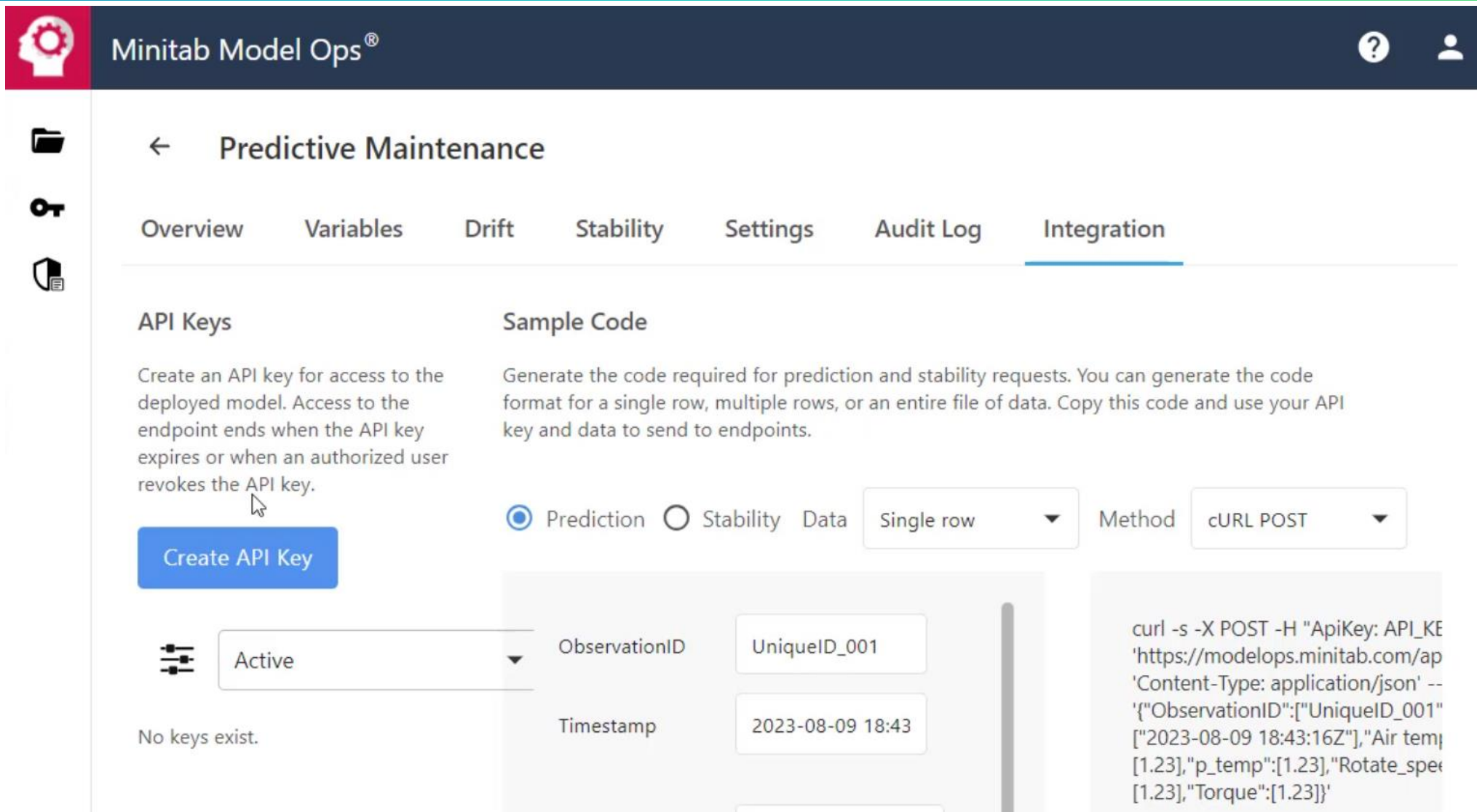


The screenshot shows the Minitab Model Ops interface. At the top, there is a header with the Minitab logo and the text "Minitab Model Ops". Below the header, there is a "Model Library" section. On the left side of the library, there are icons for a folder, a key, and a document. In the center, there are buttons for "Import a model" and "Refresh". On the right side, there is a summary: "Total models: 35 (15 deployed, 0 failed)". Below this is a table with the following columns: "Model Name", "Model Type", "Status", "Drift", and "Created". The table contains four rows of model information.

Model Name	Model Type	Status	Drift	Created
 Predictive Maint...	TreeNet® Classification	 Deployed		2023-08-09 18:34:15Z
 ACD RR_test_Au...	Discover Best Model (Cont	 Deployed		2023-08-07 15:42:44Z
 MingModelOps	Random Forests® Classific	Ready		2023-07-27 17:55:59Z
 Motors	TreeNet® Classification	 Deployed	 minimal	2023-07-07 10:17:33Z

- The model is now deployed

Step 3: Activating a Model



Minitab Model Ops®

Predictive Maintenance

Overview Variables Drift Stability Settings Audit Log **Integration**

API Keys

Create an API key for access to the deployed model. Access to the endpoint ends when the API key expires or when an authorized user revokes the API key.

[Create API Key](#)

Active

No keys exist.

Sample Code

Generate the code required for prediction and stability requests. You can generate the code format for a single row, multiple rows, or an entire file of data. Copy this code and use your API key and data to send to endpoints.

Prediction Stability Data Method

```
curl -s -X POST -H "ApiKey: API_KEY" https://modelops.minitab.com/api --header 'Content-Type: application/json' --data '{"ObservationID":["UniqueID_001"], "Timestamp": "2023-08-09 18:43:16Z", "Air temp": [1.23], "p_temp": [1.23], "Rotate_spec": [1.23], "Torque": [1.23]}'
```

Creating API Key

The screenshot shows the Minitab Model Ops interface with a 'Create API Key' dialog box open. The dialog box has a title bar 'Create API Key' and a subtitle 'Enter a key name and expiration date for your API key.' Below the subtitle is a warning message in an orange box: 'Before you distribute this API key, make sure you are authorized by your organization to publicly share the model data. Anyone with this API key has access to send prediction and stability requests for a model without additional authentication. Minitab disclaims all liability for model data that are publicly shared in violation of your organization's policies and procedures.' There are two input fields: 'Key name:' with the value 'Mikhail_Key_1' and 'Expires:' with the value '09/08/2023'. At the bottom of the dialog are 'Cancel' and 'Create' buttons. The background shows the 'API Keys' section of the interface with a 'Create API Key' button and a table with columns 'p_temp' and '1.23'.

Minitab Model Ops®

Overview

API Keys

Create an API key for a deployed model. The key is valid until it expires or when it is revoked. The API key endpoint ends with /api-key. The key is revoked when the model is updated or when the model owner revokes the API key.

Create API Key

Active

No keys exist.

Key name: Mikhail_Key_1

Expires: 09/08/2023

Cancel Create

te the code
d use your API

URL POST

POST -H "ApiKey: API_KEY" modelops.minitab.com/api --Type: application/json' -- "ationID":["UniqueID_001" 3-09 18:43:16Z"],"Air temp temp":["1.23"],"Rotate_spec rque":["1.23]'

p_temp 1.23

© 2024 Minitab, LLC

Saving API Key

The screenshot shows the Minitab Model Ops interface. A modal dialog box titled "Copy API Key" is centered on the screen. The dialog contains the following text:

Copy API Key

Copy the key now. For data security we do not store keys and you will not be able to view or copy it later.

API Key

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIzIjY3JpcHRpb25JZCI6ImQwMGFiZTdmNmQyNzQ4NmY5MGI2NzQwM2M2NmY0OGVjliwidG9rZW5JZCI6ImFiYWl0OWRkLTY4NjMtNDI5MS1iZWE1LTFiYjJmZmI1NjY4MyIsImhhdCI6MTY5MTYwNjYyMywibmJmIjoxNjknA2NjZlZCJleHAiOiJlE2OTQxMzExOTksImF1ZCI6Im1vZGVsZXliLCJpc3MiOiJtb2RlbGVyYiwic3ViljoiiYXBpliwianRpljoiZGJhMGJiOTgtMThjOS00OGE4LWI2MDctZmFhYmVjMWE5NGI3In0.2P-acDA6_8HbTnwDh0XxNK_mdFq1lBxu6b1BR8GjB7Hf4G6k14WEC2Zj4aV2MEmqwmlf5LeBZCL8NonGiJlUW42cx8toGK82P0VY1YpsqhN8fOTo2BgJxZm1BdTB_QI8RR54xS1DuTLGI9xbzJ-bXeZbKgzaKew9zJv7PQYcgFil-TvXLiKw9FAwdUwoxTpFWFvAQ1dxJVpD3ed_DDr5P-X8ZTiI98ObXu4UBAPXpYu963KZ6NOdzuKlaWtYdMJqbVS_c6ARRTX7Ywtu56S0BJIT3O4JpomqAILGC3Nq2sToyCBpjith3RtyVGszXkdmeW7bVFaR2_SHp7wk4LXSzNA
```

At the bottom of the dialog are two buttons: "Close" and "Copy to clipboard". A mouse cursor is pointing at the "Copy to clipboard" button.

In the background, the Minitab Model Ops interface is visible. The top navigation bar includes a gear icon, the text "Minitab Model Ops", a question mark icon, and a user profile icon. The left sidebar has icons for Overview, API Keys, and a document icon. The main content area shows "Overview" and "API Keys" sections. The "API Keys" section includes a description: "Create an API key for a deployed model. The key expires or when it is revoked." Below this is a "Create API Key" button and a table with columns for "Active" and "Name". One entry is visible: "Mikh..." with a sub-entry "mgolovnya...".

The Model is Active!

Minitab Model Ops[®]

?
👤

Overview
Variables
Drift
Stability
Settings
Audit Log
Integration

API Keys

Create an API key for access to the deployed model. Access to the endpoint ends when the API key expires or when an authorized user revokes the API key.

Create API Key

☰

Active

▼

Mikh...

●

acti...

⋮

active

Sample Code

Generate the code required for prediction and stability requests. You can generate the code format for a single row, multiple rows, or an entire file of data. Copy this code and use your API key and data to send to endpoints.

Prediction
 Stability
 Data

Single row

Method

cURL POST

ObservationID	UniqueID_001
Timestamp	2023-08-09 18:43
Air temp	1.23
p_temp	1.23

```
curl -s -X POST -H "ApiKey: API_KE
'https://modelops.minitab.com/ap
'Content-Type: application/json' --
'{"ObservationID":["UniqueID_001"
["2023-08-09 18:43:16Z"],"Air temp
[1.23],"p_temp":[1.23],"Rotate_spe
[1.23],"Torque":[1.23]}'
```

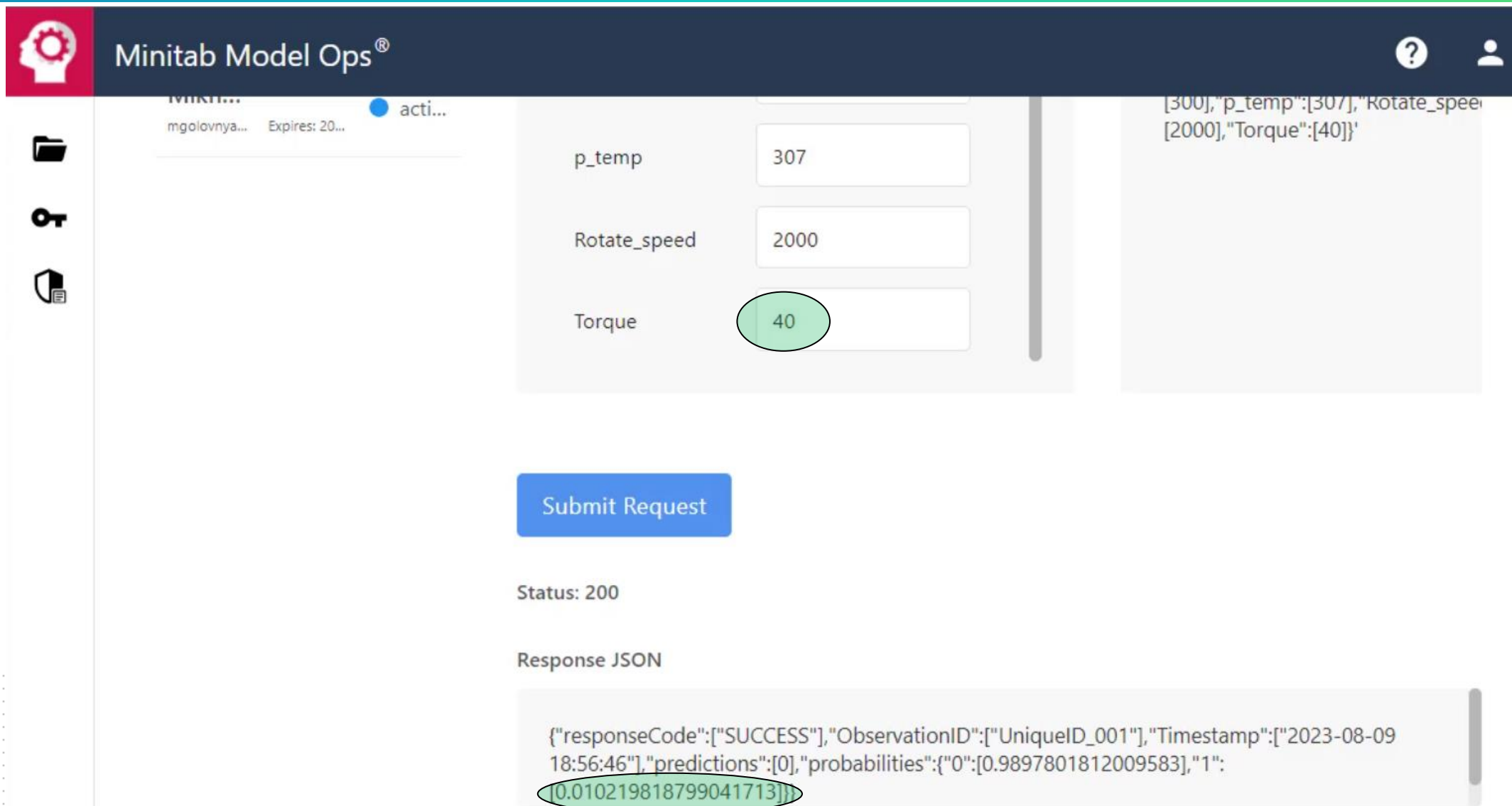

Scoring a Record Manually

The screenshot displays the Minitab Model Ops interface for manual scoring. At the top, the header includes the Minitab logo, the text "Minitab Model Ops", and navigation icons for help and user profile. Below the header, a blue button labeled "Create API Key" is visible. The main interface is divided into several sections:

- Model Information:** A dropdown menu is set to "Active". Below it, a model card for "Mikh..." (mgolovnya...) is shown with an expiration date of "Expires: 20..." and a status of "acti...".
- Input Fields:** A vertical stack of input boxes for the following parameters:
 - Timestamp: 2023-08-09 18:56
 - Air temp: 300
 - p_temp: 307
 - Rotate_speed: 2000
 - Torque: 40
- API Call:** A text area on the right contains a cURL command:

```
curl -s -X POST -H "ApiKey: API_KEY" https://modelops.minitab.com/api/v1/predictions 'Content-Type: application/json' --data '{"ObservationID":["UniqueID_001"], "Timestamp": ["2023-08-09 18:56:46Z"], "Air temp": [300], "p_temp": [307], "Rotate_speed": [2000], "Torque": [40]}'
```
- Action:** A blue "Submit Request" button is located at the bottom center, with a mouse cursor hovering over it.

Scoring a Record Manually



The screenshot displays the Minitab Model Ops interface. At the top, the header includes the Minitab logo and the text "Minitab Model Ops®". Below the header, there is a navigation sidebar with icons for home, search, and documents. The main content area shows a form for manual scoring with three input fields: "p_temp" (307), "Rotate_speed" (2000), and "Torque" (40). The "Torque" field is highlighted with a green circle. To the right of the form, a JSON response is visible, containing the same parameter values. Below the form is a blue "Submit Request" button. Underneath the button, the status is shown as "Status: 200". The "Response JSON" section displays a JSON object with a "responseCode" of "SUCCESS", an "ObservationID" of "UniqueID_001", a "Timestamp", and "predictions" and "probabilities" arrays. The "probabilities" array contains two values, with the first value, "0.010219818799041713", highlighted by a green circle.

Minitab Model Ops®

mgolovnya... Expires: 20... acti...

p_temp 307

Rotate_speed 2000

Torque 40

[300,"p_temp":[307],"Rotate_speed":[2000],"Torque":[40]]'

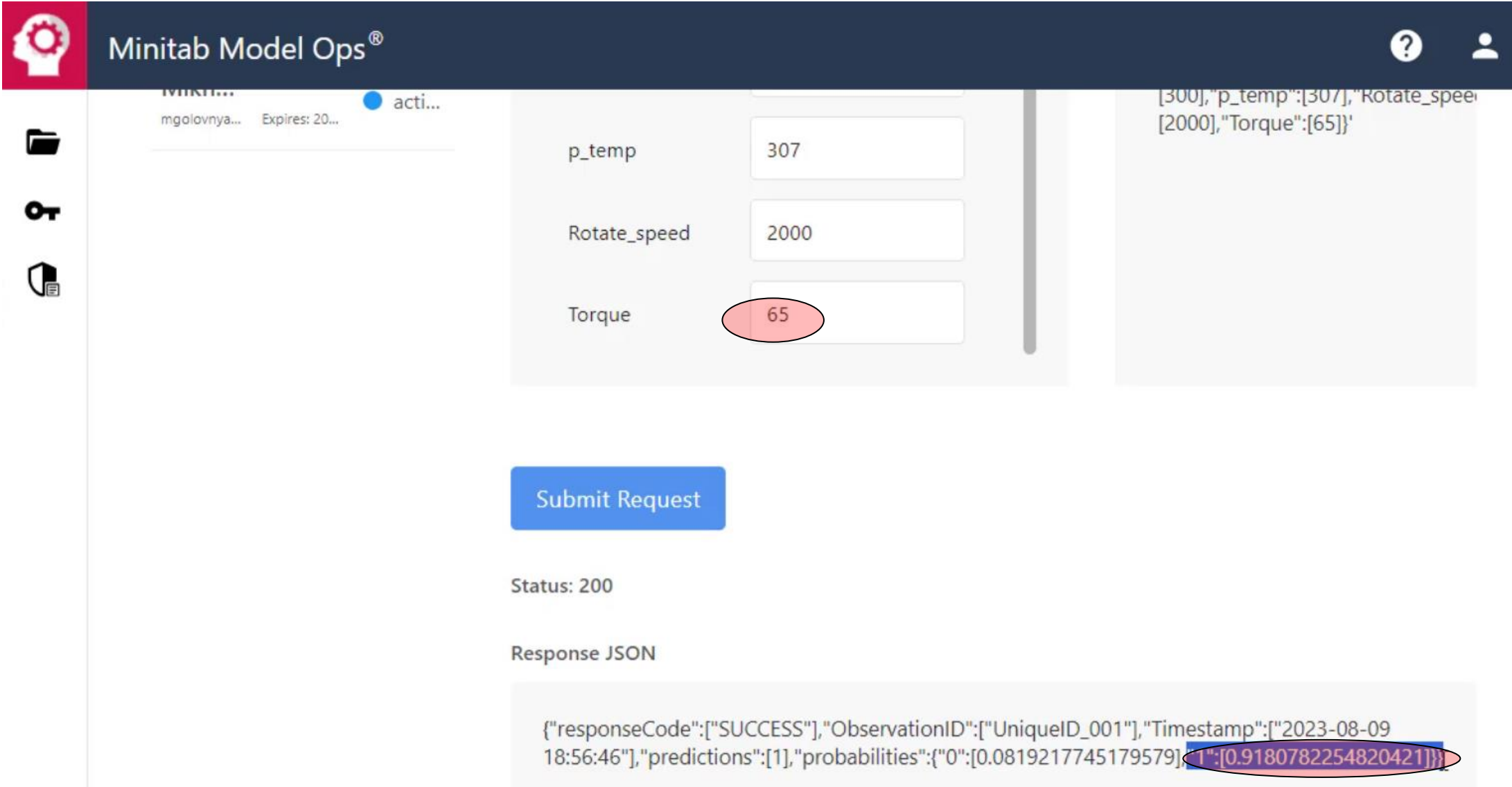
Submit Request

Status: 200

Response JSON

```
{"responseCode":["SUCCESS"],"ObservationID":["UniqueID_001"],"Timestamp":["2023-08-09 18:56:46"],"predictions":[0],"probabilities":{"0":[0.9897801812009583],"1": [0.010219818799041713]}}
```

Scoring a Different Record



The screenshot displays the Minitab Model Ops interface. At the top, the Minitab logo and 'Minitab Model Ops' are visible. Below the header, there is a navigation sidebar with icons for home, search, and documents. The main content area shows a scoring request form with three input fields: 'p_temp' (307), 'Rotate_speed' (2000), and 'Torque' (65). The 'Torque' field is highlighted with a red oval. To the right of the form, a JSON response is shown, containing a prediction of 1 and a probability of 0.9180782254820421, which is also highlighted with a red oval. Below the form is a blue 'Submit Request' button. The status is 'Status: 200' and the response JSON is displayed below.

Minitab Model Ops[®]

mgolovnya... Expires: 20... ● acti...

p_temp 307

Rotate_speed 2000

Torque 65

[300],"p_temp":[307],"Rotate_speed":[2000],"Torque":[65]}'

Submit Request

Status: 200

Response JSON

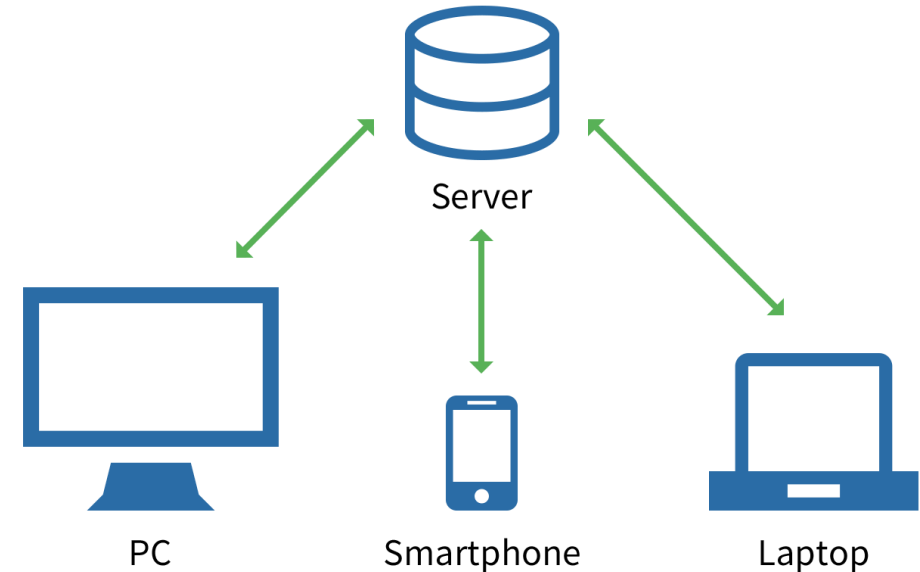
```
{"responseCode":["SUCCESS"],"ObservationID":["UniqueID_001"],"Timestamp":["2023-08-09 18:56:46"],"predictions":[1],"probabilities":{"0":[0.0819217745179579],"1":[0.9180782254820421]}}
```

Survey 3

- ▶ You need to score your data from
 - Python
 - R
 - Minitab Connect
 - Other

Universal Scoring Summary

- Reference a **model repo**
 - <https://modelops.minitab.com/api/score>
- Reference a **deployed model**
 - Unique API Key
- Post a **data record**
- Receive **the predicted response**



Scoring a Record from Python

```
In [9]: import requests
key_secret = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIzIjpbPCpHRpb25JZCI6ImQwMGFiZTdmNmQyNzQ4NmY5MGI2
headers = {"ApiKey": key_secret, "Content-Type": "application/json"}
payload = {
    'Air temp': ['300'],
    'p_temp': ['307'],
    'Rotate_speed': ['2000'],
    'Torque': ['40']}
url = "https://modelops.minitab.com/api/score"
r = requests.post(url, json=payload, headers=headers)
print('Response Code:', r.status_code)
print('Response Content:\n', r.text)
```

Response Code: 200

Response Content:

```
{ "responseCode": ["SUCCESS"], "Timestamp": ["2023-08-09 19:33
{"0": [0.9897801812009583], "1": [0.010219818799041713]}]}
```



Scoring a Different Record from Python

```
In [10]: import requests
key_secret = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIzIjpcHRpb25JZCI6ImQwMGFiZTdmNmQyNzQ4NmY5MGI2
headers = {"ApiKey": key_secret, "Content-Type": "application/json"}
payload = {
    'Air temp': ['300'],
    'p_temp': ['307'],
    'Rotate_speed': ['2000'],
    'Torque': ['65']}
url = "https://modelops.minitab.com/api/score"
r = requests.post(url, json=payload, headers=headers)
print('Response Code:', r.status_code)
print('Response Content:\n', r.text)
```

Response Code: 200

Response Content:

```
{ "responseCode": ["SUCCESS"], "Timestamp": ["2023-08-09 1
{"0": [0.0819217745179579], "1": [0.9180782254820421]} }
```



Scoring a Dataset from Python

```
In [ ]: # Sending a batch of rows for scoring / drift

import pandas
import json
import requests

# Read/prep the original data
m_data = pandas.read_csv("predictive_maintenance2_subset1.csv")
headers = list(["ObservationID", "Air_temp", "p temp", "Rotate_speed", "Torque", "Machine_failure"])
m_data = m_data[m_data.columns.intersection(headers)]

# Format the request in Python
key_secret = "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIzIjpcHRpb25JZCI6ImQwMGFiZTdmNmQyNzQ4NmY5MGI2NzQwM2M2NmY0OGVjIiwidG9rZmVudCI6ImVudG9rZmVudCJ9"
scoreHeader = {'ApiKey': key_secret, 'Content-Type': 'application/json'}
test=m_data.to_dict(orient="list")

# Send the batch for scoring to Model Ops
scoringURL = "https://modelops.minitab.com/api/score"
post_scores=requests.post(scoringURL,headers=scoreHeader,data=json.dumps(test))

# Extract the results of scoring
dataFromModelOps = json.loads(post_scores.content.decode("utf-8"))
predictionsFromModelOps = dataFromModelOps['predictions'] #raw scored values
```


Conclusion

- **TreeNet** can be used to suggest classical **formula-based deployments** of ML models
- **MSS Model Ops** offers powerful **enterprise-level solution** for model deployment and monitoring
- Once deployed in Model Ops, the model can be accessed for scoring in a variety of ways
 - Web access
 - Python access
 - MSS Connect access
- But what about model monitoring?



Q&A



Upcoming In-Person Events

Dates and Location in the US

- Rosemont, IL – June 18th
- Columbus, OH – August 15th
- Dallas, TX – September 10th
- Anaheim, CA – October 10th



Minitab 
EXCHANGE

You have data. We have solutions. Imagine the possibilities.

At Minitab, we help customers around the world leverage the power of data analysis to gain insights and make a significant impact on their organizations. By unlocking the value of data, Minitab enables organizations to improve performance, develop life changing innovations and meet their commitments of delivering high quality products and services and outstanding customer satisfaction.



thank you

Gracias	ευχαριστώ	Danke	Grazie	благодаря
Hvala	Obrigado	Kiitos	شكراً	Tak
Ahsante	Teşekkürler	متشكراً	Salamat Po	감사합니다
Cám ơn	شكريه	Terima Kasih	Dank u Wel	Děkuji
நன்றி	Köszönöm	ありがとう ございます	ขอขอบคุณครับ	Dziękuję
谢谢	Tack	Mulțumesc	спасибо	Merci
תודה	多謝晒	дядкую	Ďakujem	