HACKtheMACHINE NEW YORK

TRACK 2: Data Science: Cleared for Takeoff

Naval aircraft are complex systems that transmit, receive, and collect gigabytes of data during every mission. Normally when a system fails, it warns the aircrew with flashing lights and audible alarms or at least registers a fault code in the onboard computer system. However, sometimes pilots experience unpredictable behavior without clear warnings, or the fault codes don't align to what the pilot experienced. Even more frustrating is when the problems are intermittent. Maintainers can spend hours pulling apart the assemblies and running through the troubleshooting procedures to try and determine the root cause, but they can't replicate the fault or find an issue. The issue is reported as Malfunction Code "799" or "No Fault Found" in the maintenance log.

Often, a problem does indeed exist. Issues with corrosion or wiring can cause fault codes that are seemingly unrelated to the problems the aircraft is experiencing. Fortunately, sometimes the story is in the data. Patterns in the aircraft fault codes and maintenance logs can reveal when wires within the aircraft began to chafe or corrosion causes systems on the aircraft to intermittently report errors before they can be detected by the pilot or the maintainer. Finding these unique patterns in the data could reveal the damage before an incident occurs and save dollars and potentially lives.

Data Intro:

For this challenge we are providing you with two unique datasets that have never before been publicly available. The first dataset is the Maintenance Action Form (MAF) data, which contains a record of maintenance actions performed on the aircraft. It is a record of everything maintainers have done to the aircraft including: routine inspections, part removal and replacement, mission-related changes, and unscheduled maintenance, including unforeseen failures. The second dataset is the Memory Unit (MU) data. The MU data is a time-stamped record of all failures or anomalies automatically reported by the aircraft's avionics system, similar to an error log your auto mechanic might download from your car when the check engine light comes on.

Challenge Intro:

The challenge is to identify patterns and correlations in the MU data and determine how those relate to the maintenance actions. For example, a maintainer accidently switches two wires during a routine inspection. From that point on, three seemingly unrelated fault codes appear with extremely high correlation. The wiring mismatch is fixed two months later, but one intermittent fault code remains. Upon further inspection, it is determined that this fault code is caused by chafing in a wiring harness. Once the harness is repaired the fault code ceases. We need your help discovering patterns of aircraft generated fault codes that are indicative "signature" conditions related to corrosion and wiring issues. The early identification of these problems will help reduce the downtime and number of mishaps experienced by F-18s.

CHALLENGE

You have ~7 years of data from 45 F-18s. The dataset contains 13 aircraft that were subject to a special set of inspections to fix wiring issues. The wiring issues were corrected at some point in the past. The remaining aircraft have never before been analyzed. Your task is to develop analytic techniques to identify patterns of interest in the MSP codes (MU data) and relate them to the maintenance actions (MAF data). Relevant analytical techniques may include timeseries, correlation analysis, and clustering algorithms. In particular, we are interested in corrosion-related problems and have tagged the data with a flag for corrosion-related maintenance actions.

Challenge 1: Up to **40 Points** will be awarded for your ability to identify the MSP code patterns in the 13 aircraft with known wiring issues. For this challenge, your objective is to identify which 13 of the 45 aircraft had the wiring inspections and subsequent repairs and to provide a list of the maintenance actions that you believe fixed the wiring issues. You may provide up to 200 maintenance actions as part of your solution. Additionally, you can provide up to 10 MSP codes for each maintenance action that you believe are associated with the wiring fixes.

Hint #1: Intermittent MSP codes related to the issue likey stopped after the repair.

Hint #2: Some aircraft have clusters of wiring related MAFs and the same MSP codes can be correctly attributed to more than one MAF.

Hint #3: Some aircraft had more than one wiring inspection.

There is a challenge 1 submission spreadsheet template that will be provided. In order to receive points, the algorithms and methodology behind your solution must be provided and the evaluation criteria will include accuracy of your solution as well as generalizability to other non-wiring issues.

Challenge 2: Up to **50 Points** will be awarded for algorithms, techniques, and visualizations that most effectively identify demonstrate patterns in the relationship between MSP codes and maintenance actions, particularly those related to corrosion. Variables of interest include: Aircraft, MSP Code(s) (with start and stop dates), correlation with maintenance actions or with other MSP codes, frequency, duration, and flight mode. These patterns could be within aircraft or across aircraft and might correspond to aircraft sub-type, squadron, or other variables.

For individual aircraft, the Navy has anecdotal examples of a corrosion-related maintenance actions resulting in intermittent MSP codes stopping. Across aircraft, the Navy has discovered that sometimes when an aircraft is moved between certain squadrons the transfer inspection results in intermittent MSP codes stopping. If you are able to uncover a systematic way to discover these features in the data, it may change the way maintenance is performed.

Your solution should include a visualization component, as the ability to display data is a critical component in recognizing patterns of fault codes. Points will be given for MSP code patterns and the corresponding MAFs that the Navy should investigate, as well as the development of a data visualization tool that clearly communicates the fault code patterns to the aviation enterprise. Points will be awarded for informative approaches to pattern recognition as evaluated by subject matter experts. The criteria will include technical strength of the methodology, generalizability, creativity/uniqueness of the approach, and the degree to which the visualizations provide insight and support your conclusions.

Containerization: up to **10 Points** will be given to teams that containerize their algorithms to run within on the cloud-based Agile Core Services (ACS) environment (see Environment and Tools section). These algorithms may be run on the ACS system after the hackathon. We will have staff on hand to assist if you've never done this before.

Environment Review: Up to **5 Bonus Points** will be awarded for the feedback on the ACS environment. Specifically, we will be looking for feedback on usability and tools. Regarding tools, we would like your thoughts on the current tool suite as well as tools that should be included to support data science. A feedback template will be provided as part of the participant materials.

DATA

The data is from 2012-2019 and is available in csv format and contains the fields below.

Memory Unit (MU) Data Fields

AIRCRAFT	The unique identifier for the aircraft, maps to Aircraft in the MAF data
SQUADRON	The grouping the aircraft is currently assigned to. This changes over time as aircraft are moved around.
LOT	The manufacturing grouping for the aircraft. Lots typically consist of 12+ aircraft.
MSP	Maintenance Service Panel code, similar to the standard fault codes you can download from an automobile.
ZULU_TIME	GMT time record of the fault (Note: The MAF data is in local time, so the time stamps will not correlate exactly)
FLIGHT_MODE	What the aircraft was doing when the fault code happened. (Repeated patterns of a fault code while the aircraft is on the ground can mean a maintainer is servicing the aircraft or trying to replicate a fault.)

Maintenance Action Form Data Fields

Job Code	A unique identifier assigned to each maintenance task.
Aircraft	The identifier for the aircraft, maps to AIRCRAFT in the MU data
Transaction Code	The type of data being reported. Descriptions are on page E-29 of Maintenance Documentation Codes
Malfunction Code	Codes that reference the type of malfunction (failure vs non-failure) and where it occurred. 799 is no fault found and may be of particular interest when paired with the MU data.

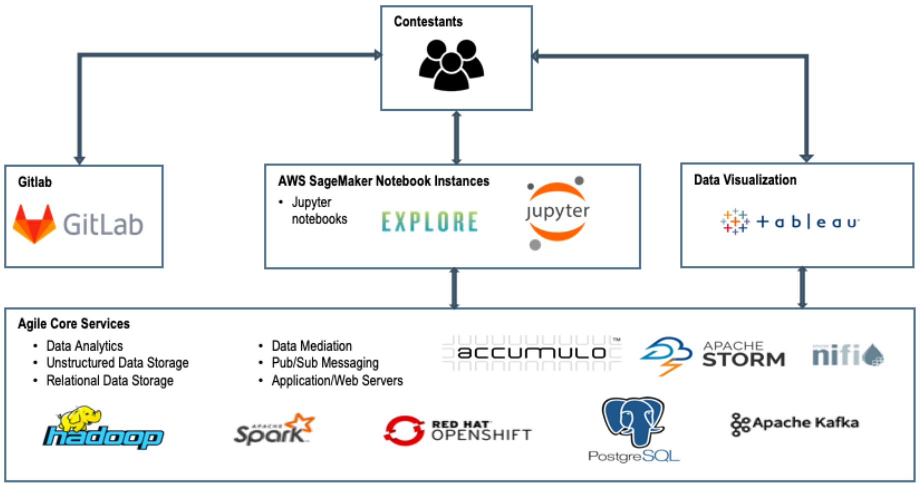
Action Taken Code	The type of repair. These codes are frequently misclassified. Descriptions are on page E-1 of Maintenance Documentation Codes.
Description of Problem	A description of what needed to be fixed on the aircraft.
Correction of Problem	A description of how the problem on the aircraft was corrected.
Received Date	The date the maintenance action was received in local time
Completion Date	The date the maintenance action was completed in local time.
Corrosion	A yes flag that indicates whether the maintenance action was corrosion-related.
Bare Metal	A yes flag that indicates whether the maintenance action references bare metal, which can be indicative of corrosion.
Preventative Corrosion Treatment	A yes flag that indicates whether the maintenance action was preventative. Preventative maintenance is often related to corrosion.
Routine	A yes flag that indicates whether the maintenance action was routine.
Unscheduled	A yes flag that indicates whether the maintenance action was unscheduled.
Mission-Related Maintenance	A yes flag that indicates whether the maintenance action was related to mission needs, such as adding a particular sensor or capability.
Failure	A yes flag that indicates whether the maintenance action was related to a failure.

ENVIRONMENT AND TOOLS

Please bring your own computer that you will use extensively during the challenge. There
Please bring your own computer that you will use extensively during the challenge. There are no restrictions on what types of data science tools or operating system you can use, but we will be optimizing for standard Windows and Mac-based systems. Please ensure that you have full permissions on the machine that you choose to use and that at least some members of your team can install software as needed.

For this challenge we will be providing contestants an inside look at a realistic Navy environment called Agile Core Services (ACS). ACS is part of the Navy's Consolidated Afloat Networks and Enterprise Services (CANES). As ACS is still being developed, participants in this challenge have the opportunity to shape the future of how this environment is developed. The diagram below depicts how the environment has been set up for this hackathon and how participants can plan to interact with the tools provided.

As the diagram above shows, each team will be assigned a GitLab Repository. This will be your team's home base during the challenge. Within this repository there will be guides, several Jupyter notebooks that have been set up to enable your team to get started on the challenge as quickly as possible, and where you can plan to upload content for your final submissions. Access into these repositories will be given on the first day of the competition.



A key to success during the hackathon will be effectively using the tools available. If you want to brush up on your skills around any of the tools we've assembled a list of tutorials and quick start guides below.

- **Jupyter Notebook Tutorial.** A standard in the data scientists toolbelt, you should be comfortable using Jupyter notebooks to analyze data and create algorithms.
- **Amazon Sagemaker Tutorial.** This tool will be available as a platform for data analysis and machine learning.
- **GitLab Command Line Workflow.** During the hackathon there will be a basic GitLab web UI, but participants can use command line workflows for full functionality.
- **Tableau Desktop Tutorial.** Each team will have access to a Tableau account that can be used to create data visualization.

JUDGING AND PRIZES

Judging will occur both during and after the HACKtheMACHINE event. Your team will provide a short presentation of results to a judging panel on Sunday afternoon. Your team will also be expected to provide the code and methods that were used to develop your solution. Entries will be judged by a team of experts and the criteria will include accuracy of the solution, technical strength of the methodology, generalizability, creativity/uniqueness of the approach, the degree to which the visualizations provide insight and support your conclusions, and your use of the challenge environment.

Winners will be announced via the website no later than Sept 30th. Cash prizes will be awarded to the winners in accordance with the prize guidelines, which can be found here.

If you have additional questions that aren't answered here, please check out the Event FAQ or reach out to us at HACKtheMACHINE@fathom5.co