# ![](https://ga-dash.s3.amazonaws.com/production/assets/logo-9f88ae6c9c3871690e33280fcf557f33.png) Kaggle Competition - Starter

## Introduction

Welcome to your first week of work at the Disease And Treatment Agency, division of Societal Cures In Epidemiology and New Creative Engineering (DATA-SCIENCE). Time to get to work!

Due to the recent epidemic of West Nile Virus in the Windy City, we've had the Department of Public Health set up a surveillance and control system. We're hoping it will let us learn something from the mosquito population as we collect data over time. Pesticides are a necessary evil in the fight for public health and safety, not to mention expensive! We need to derive an effective plan to deploy pesticides throughout the city, and that is \*\*exactly\*\* where you come in!

## Dataset

The dataset, along with description, can be found here: [https://www.kaggle.com/c/predict-west-nile-virus/](https://www.kaggle.com/c/predict-west-nile-virus/).

\*\*This is also where you will be submitting your code for evaluation\*\*. We will be using the Kaggle Leaderboard to keep track of your score. The public leaderboard uses roughly 30% of the dataset to score an AUC (Area Under Curve) metric. [You can read more about the scoring metric here](https://www.kaggle.com/wiki/AreaUnderCurve).

> If you do not already have a Kaggle account, you will need to sign up on the website. Also note that you will be submitting a "Late Submission" on Kaggle because the official competition has ended. You can use the leaderboard to see how your results compare against roughly 1300 other data science teams!

You can submit predictions as many times as you want to Kaggle, but there is a limit of 5 submissions per day. Be intentional with your submissions!

#### Navigating Group Work

This project will be executed as a group. To make your team as effective and efficient as possible you should do the create a shared GitHub repo and project planning document as described in the deliverables section below.

## Deliverables

\*\*GitHub Repo\*\*

1. Create a GitHub repository for the group. Each member should be added as a contributor.

2. Retrieve the dataset and upload it into a directory named `assets`.

3. Generate a .py or .ipynb file that imports the available data.

\*\*Project Planning\*\*

1. Define your deliverable - what is the end result?

2. Break that deliverable up into its components, and then go further down the rabbit hole until you have actionable items. Document these using a project managment tool to track things getting done. The tool you use is up to you; it could be Trello, a spreadsheet, GitHub issues, etc.

3. Begin deciding priorities for each task. These are subject to change, but it's good to get an initial consensus. Order these priorities however you would like.

4. You planning documentation (or a link to it) should be included in your GitHub repo.

\*\*EDA\*\*

1. Describe the data. What does it represent? What types are present? What does each data points' distribution look like? Discuss these questions, and your own, with your partners. Document your conclusions.

2. What kind of cleaning is needed? Document any potential issues that will need to be resolved.

\*\*Note:\*\* As you know, EDA is the single most important part of data science. This is where you should be spending most of your time. Knowing your data, and understanding the status of its integrity, is what makes or breaks a project.

\*\*Modeling\*\*

1. The goal is of course to build a model and make predictions that the city of Chicago can use when it decides where to spray pesticides! Your team should have a clean Jupyter Notebook that shows your EDA process, your modeling and predictions.

2. Conduct a cost-benefit analysis. This should include annual cost projections for various levels of pesticide coverage (cost) and the effect of these various levels of pesticide coverage (benefit). \*(Hint: How would we quantify the benefit of pesticide spraying? To get "maximum benefit," what does that look like and how much does that cost? What if we cover less and therefore get a lower level of benefit?)\*

3. Your final submission CSV should be in your GitHub repo.

\*\*Presentation\*\*

\* Audience: You are presenting to members of the CDC. Some members of the audience will be biostatisticians and epidemiologists who will understand your models and metrics and will want more information. Others will be decision-makers, focusing almost exclusively on your cost-benefit analysis. Your job is to convince both groups of the best course of action in the same meeting and be able to answer questions that either group may ask.

\* The length of your presentation should be about 10 minutes (a rough guideline: 1 minute intro, 5 minutes on model, 3 minutes on cost-benefit analysis, 1 minute recommendations/conclusion). Touch base with your local instructor... er, manager... for specific logistic requirements!

---

\*\*Your project is due at 9:00 AM SGT on Friday, 29th Jan 2021.\*\*

---

### Project Feedback + Evaluation

For all projects, students will be evaluated on a simple 4 point scale (0-3 inclusive). Instructors will use this rubric when scoring student performance on each of the core project requirements:

Score | Expectations

----- | ------------

\*\*0\*\* | \_Does not meet expectations. Try again.\_

\*\*1\*\* | \_Approaching expectations. Getting there...\_

\*\*2\*\* | \_Meets expectations. Great job.\_

\*\*3\*\* | \_Surpasses expectations. Brilliant!\_

### Rubric

Your final assessment ("grade" if you will) will be calculated based on a topical rubric (see below). For each category, you will receive a score of 0-3. From the rubric you can see descriptions of each score and what is needed to attain those scores.

For Project 3 the evaluation categories are as follows:

- [Organization](#organization)

- [Data Structures](#data-structures)

- [Python Syntax and Control Flow](#python-syntax-and-control-flow)

- [Probability and Statistics](#probability-and-statistics)

- [Modeling](#modeling)

- [Presentation](#presentation)

#### Organization

Clearly commented, annotated and sectioned Jupyter notebook or Python script. Comments and annotations add clarity, explanation and intent to the work. Notebook is well-structured with title, author and sections. Assumptions are stated and justified.

| Score | Status | Examples |

|-------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 0 | Does not Meet Expectations | 1. Comments and annotations are \*\*absent\*\* <br> 2. There is no clear notebook structure <br> 3. Assumptions are not stated |

| 1 | Approaching Expectations | 1. Comments are present but generally unclear or uninformative (e.g., comments do not clarify, explain or interpret the code) <br> 2. There are some structural components like section/subsection headings <br> 3. Assumptions are stated but not justified |

| 2 | Meets Expectations | 1. Comments and annotations are clear and informative <br> 2. There is a clear structure to the notebook with title and appropriate sectioning <br> 3. Assumptions are both stated and justified |

| 3 | Exceeds Expectations | 1. Comments and annotations are clear, informative and insightful <br> 2. There is a helpful and cogent structure to the notebook that clarifies the analysis flow <br> 3. Assumptions are stated, justified and backed by evidence or insight |

#### Data Structures

Python data structures including lists, dictionaries and imported structures (e.g. DataFrames), are created and used correctly. The appropriate data structures are used in context. Data structures are created and accessed using appropriate mechanisms such as comprehensions, slices, filters and copies.

| Score | Status | Examples |

|-------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 0 | Does not Meet Expectations | 1. Appropriate data structures are not identified or implemented <br> 2. Data structures are not created appropriately <br> 3. Data structures are not accessed or used effectively |

| 1 | Approaching Expectations | 1. Contextually appropriate data structures are identified in some but not all instances <br> 2. Data structures are created successfully but lacked efficiency or generality (e.g., structures were hard-coded with values that limits generalization; brute-force vs automatic creation/population of data) <br> 3. Data structures are accessed or used but best practices are not adopted |

| 2 | Meets Expectations | 1. Contextually appropriate data structures are identified and implemented given the context of the problem <br> 2. Data structures are created in an effective manner <br> 3. Data structures are accessed and used following general programming and Pythonic best practices |

| 3 | Exceeds Expectations | 1. Use or creation of data structures is clever and insightful <br> 2. Data structures are created in a way that reveals significant Pythonic understanding <br> 3. Data structures are used or applied in clever or insightful ways |

#### Python Syntax and Control Flow

Python code is written correctly and follows standard style guidelines and best practices. There are no runtime errors. The code is expressive while being reasonably concise.

| Score | Status | Examples |

|-------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 0 | Does not Meet Expectations | 1. Code has systemic syntactical issues <br> 2. Code generates incorrect results <br> 3. Code is disorganized and needlessly difficult |

| 1 | Approaching Expectations | 1. Code is generally correct with some runtime errors <br> 2. Code logic is generally correct but does not produce the desired outcome <br> 3. Code is somewhat organized and follows some stylistic conventions |

| 2 | Meets Expectations | 1. Code is syntactically correct (no runtime errors) <br> 2. Code generates desired results (logically correct) <br> 3. Code follows general best practices and style guidelines |

| 3 | Exceeds Expectations | 1. Code adopts clever or advanced syntax <br> 2. Code generates desired results in an easily consumable manner (e.g., results are written to screen, file, pipeline, etc, as appropriate within the flow of the analysis) <br> 3. Code is exceptionally expressive, well formed and organized |

#### Probability and Statistics

Descriptive and inferential statistics are calculated and applied where appropriate. Probabilistic reasoning is demonstrated. There is a clear understanding of how probability and statistics affects the analysis being performed.

| Score | Status | Examples |

|-------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 0 | Does not Meet Expectations | 1. Descriptive statistical calculations are absent <br> 2. Inferential statistical calculations are absent <br> 3. Probabilities or statistics are not relevant given the context of the analysis |

| 1 | Approaching Expectations | 1. Descriptive statistics are present in some cases <br> 2. Inferential statistics are present in some cases <br> 3. Probabilities or statistics are somewhat relevant to the analysis context |

| 2 | Meets Expectations | 1. Descriptive statistics are calculated in all relevant situations <br> 2. Inferential statistics are calculated in all relevant situations <br> 3. Probabilities or statistics are relevant to the analysis |

| 3 | Exceeds Expectations | 1. Descriptive statistics are calculated, interpreted and visualized (where appropriate) <br> 2. Inferential statistics are calculated, interpreted and visualized (where appropriate) <br> 3. Probabilities or statistics are leveraged to draw meaningful or insightful conclusions |

#### Modeling

Data is appropriately prepared for modeling. Model choice matches the context of the data and the analysis. Model hyperparameters are optimized. Model evaluation is robust. Model results are extracted and explained either visually, numerically or narratively.

| Score | Status | Examples |

|-------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 0 | Does not Meet Expectations | 1. Data is not prepared for modeling.<br>2. Models are not implemented or not implemented fully.<br>3. Model hyperparameters are not considered.<br>4. Model evaluation is not performed.<br>5. Model results are unavailable or not extracted. |

| 1 | Approaching Expectations | 1. Data has some null values, inappropriate types and/or improper handling of categorical labels.<br>2. Model choice is questionable given the objective of the analysis.<br>3. Model hyperparameters are insufficiently or not optimized.<br>4. Model evaluation is performed but the evaluation is not generalizable.<br>5. Model results are extracted but not explained or interpreted. |

| 2 | Meets Expectations | 1. Data is free from nulls and correctly typed for the given model.<br>2. Model choice is appropriate to the analysis.<br>3. Model hyperparameters are optimally selected.<br>4. Model evaluation reflects generalizeable performance.<br>5. Model results are extracted and explained either visually, numerically or naratively. |

| 3 | Exceeds Expectations | 1. Data is pristinely prepared with creative or useful feature engineering.<br>2. Model selection is justified and demonstrates an awareness of tradeoffs.<br>3. Model hyperparameters are optimized and the optimization is demonstrated/justified.<br>4. Model evaluation reflects generalizable performance and is interpreted in the context of the analysis.<br>5. Model results are explained, interpreted and related to the overarching analysis goals. |

#### Presentation

The goal, methodology and results of your work are presented in a clear, concise and thorough manner. The presentation is appropriate for the specified audience, and includes relevant and enlightening visual aides as appropriate.

| Score | Status | Examples |

|-------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 0 | Does not Meet Expectations | 1. The problem was not well explained or ambiguous. <br> 2. The level of technicality was far above or below the target audience. <br> 3. The presentation went substantially over or under time. <br> 4. The speaker's voice was difficult to hear of unclear. <br> 5. The presentation visuals did not seem to support the talk. |

| 1 | Approaching Expectations | 1. The problem was stated but was not 100% clear. <br> 2. The level of technicality was was good at times, but too low or too high at other times given the target audience. <br> 3. The presentation was given went slightly over or under time. <br> 4. The speaker's voice was at times difficult to understand. <br> 5. The presentation visuals were generally helpful, but some of them were either too complex or disconnected from the narrative. |

| 2 | Meets Expectations | 1. The problem was framed appropriately for the audience. <br> 2. The level of technicality was appropriate to the target audience. <br> 3. The presentation was given within the allocated timeframe. <br> 4. The speaker's voice had volume and clarity. <br> 5. The presentation visuals were helpful and supportive. |

| 3 | Exceeds Expectations | 1. The problem was expertly stated and compelling. <br> 2. The level of technicality was perfect for the target audience. <br> 3. The presentation was given within the allocated timeframe and paced evenly throughout. <br> 4. The speaker's voice was clear, understandable and consistent. <br> 5. The presentation visuals provided distinct insight, supported the speaker from the background, and were not distracting. |