# ECE471 Fall 2022
# Project 1: Safety Analysis of Unmanned Vehicles

As we discussed in class, artificial intelligence (AI)-driven technologies are being integrated into many activities that we take for granted. For example, the continued development and commercial deployment of unmanned vehicles promise a global revolution in transportation infrastructure. In this project, you will investigate one type of unmanned vehicle: Autonomous Vehicles (AVs) running on the road. Specifically, you will first learn to generate data with different scenarios and different driving conditions from an open-source simulator for autonomous driving research, Carla. By analyzing the simulated driving data, you will derive insights into the AV's safety and common causes of failure. Then we will apply our analysis to a real dataset from the California DMV to make more practical conclusions. Such insights are instrumental in developing more reliable unmanned vehicles and reducing the chance of accidents.

In this project, you will study the simulated & real-world data of AVs using fundamental concepts from probability and statistics. The concepts you will learn and apply include the following

1. Parsing and visualizing datasets (Importing, extracting, and summarizing features)
2. Basic statistical analysis of dataset
3. Probabilistic analysis of the data using concepts from ECE 313 (e.g., Probability, Conditional Probability, Total Probability)
4. Using Bayesian inference to derive posterior probabilities
5. Develop and perform statistical inference with a Bayesian Network

## Part 1: Simulation with Carla

**Background:**

Autonomous Vehicles are complex systems that use artificial intelligence (AI) and machine learning (ML) to integrate mechanical, electronic, and computing technologies to make real-time driving decisions. Several states in the USA (e.g. California, Texas, Nevada, Pennsylvania, and Florida) and other parts of the world (e.g. China [1]) have already started field-testing AVs on public roads. As AVs have started interacting more directly with humans on public roads, the safety and resilience of AVs is a significant concern (Uber's [2] fatal accident, Tesla's [3] autopilot flaw) and must be thoroughly evaluated through the analysis of data obtained during both simulations and real-world field-testing.

Simulation plays a crucial role in verifying the AV's functionality and safety in the early phase of the development, as it enables developers to virtually test millions of driving situations in order to validate the vehicle's high-fidelity sensors, actuators, LIDARs, and cameras. Simulation can even drive the development and testing of complex automated driving control software [4]. In this part, you will use one of the most popular open-source simulators for autonomous vehicles, Carla [5], which supports the development, training, and validation of autonomous driving systems, to validate a pre-trained autonomous driving agent's behavior under different driving conditions. You will run simulations of different traffic scenarios, and collect and analyze the simulation data.

**Dataset description:**

When you run the simulation with the provided scripts, the simulation results will be saved in the 'campaign_result' folder automatically. Each scenario + weather condition will have a separate folder, containing 5 files: 'route_highway.txt', '*_ctl.csv', '*_cvip.csv', '*_traj.csv', and 'run.done'. Please find the description of each file below.

**route_highway.txt**
This file stores the statistics of the simulation in a JSON-like format. The simulation stops when (1) an accident happens, (2) the time-out is triggered, or (3) the goal is reached. The simulation is "complete" only when the goal is reached, and otherwise (for condition (1) and (2)) the simulation is "incomplete". For complete simulations the "status" will be "Completed", and for incomplete simulations the "status" will be "Failed". For simulations with accident the "accident"-related fields specify what kind of accidents had happened. For example, an non-empty "collisions vehicles" indicates an accident with another vehicle.

Note that a complete simulation can still have safety-violations (infractions), indicated in the "infractions" section. If there exists an infraction (including accident), the "score composed" will be less than 100.0, otherwise if there is no infraction the "score composed" equals 100.0.

***_ctl.csv**
This file stores the values of control variables during the simulation.

| **Column Name** | **Explanation** |
| --- | --- |
| ts | Timestamp |

| | |
|---|---|
| agent_id | ID of the AV |
| throttle | The acceleration of the vehicle |
| steer | The direction of the vehicle |
| brake | Triggering the brake will slow down/stop the vehicle |

***_cvip.csv**
This file stores the values of distance from the eagle vehicle to the AV.

| **Column Name** | **Explanation** |
|---|---|
| ts | Timestamp |
| agent_id | ID of the AV |
| cvip | The distance from the eagle vehicle to the AV |

***_traj.csv**
This file stores the absolute world coordinate (telemetry) of the AV.

| **Column Name** | **Explanation** |
|---|---|
| ts | Timestamp |
| agent_id | ID of the AV |
| x | The horizontal location (left/right) of the vehicle |
| y | The moving direction of the vehicle |
| v | The speed of the vehicle along the y axis |

**run.done**
Please ignore this file.

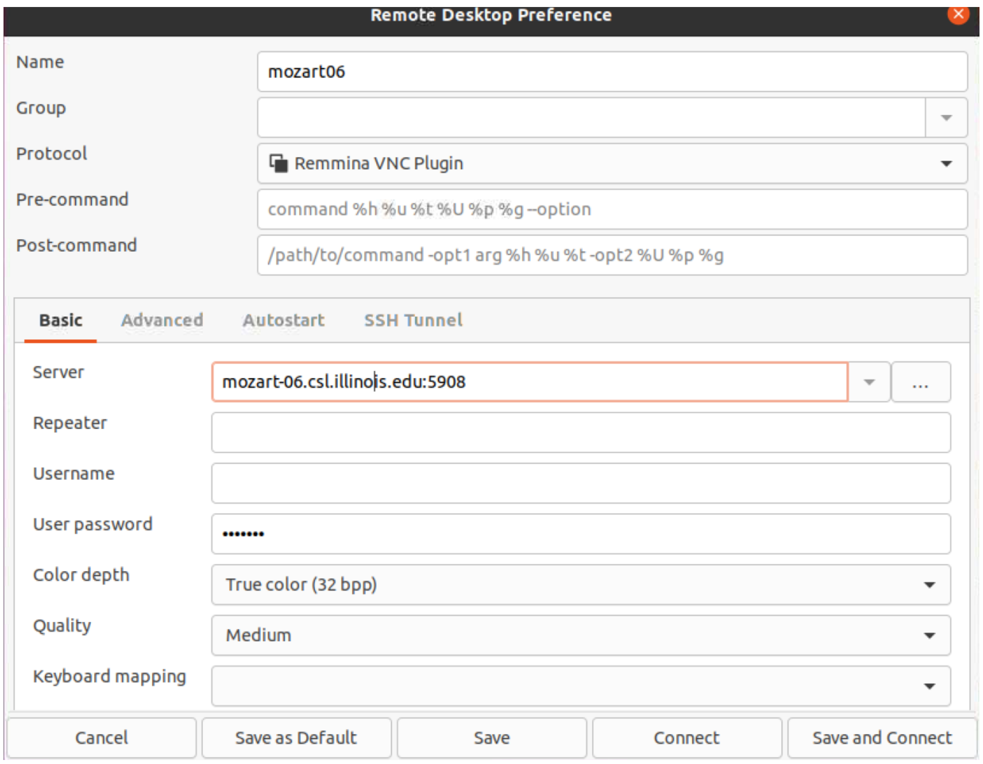
<u>Task 0 – Collecting simulated data with Carla (3 points)</u>

The Carla simulator uses a server-client structure. We have already prepared a Carla server for you, and your task is to run a python client that sets up the simulation and generates data for further analysis.

Each group will be given two accounts on the class server: a shared VNC account to view the graphic output of the Carla server, and a private group account to run the python client. You need to be in the university network (work on campus or connect to the university VPN) to gain access to the class server.

## Connecting to the VNC server

To connect to the VNC server, you will need the following information: server address, port number, and an 8-digit password, all of which will be provided to you once all the groups are formed. You will also need to install a VNC client. Here is an example setting using Remmina, a remote desktop client preinstalled in Ubuntu:




You can choose any VNC client you prefer, and adjust the color depth and quality based on your network condition.

Once you are connected to the VNC desktop, a Carla server will already be opened for you. Please do not do anything else other than playing with the Carla server in the VNC Desktop, including starting any other programs or trying to reboot the machine. Violators will lose access to the server. If you accidentally close the Carla server, a background script will automatically restart the server.

Please be aware that the VNC server is shared among 3-4 groups. Before you sign in, make sure to reserve your time on the spreadsheet shared among the groups sharing the same account.

## Running the python client

You should run the python client in your private group account. Do not work on this part in the VNC desktop. You can connect to the group account using "***ssh account_name@server_address***". Copy the MP1 folder into your account with ***scp***.
Before running the main script, open *env.sh* in the MP1 folder with your preferred text editor (you need to enable X forwarding to run a graphic text editor), and make sure that the port number(the carla server port, not the VNC server port) and display number matches the ones that are assigned to your group. If you use vim, press i to enable editing, and enter ***:wq*** to save the change.

After saving the change, run ***"source env.sh"*** to set up the environment variables (you need to run it every time you start a new shell). This script will activate a shared anaconda environment, and direct all graphic outputs to the VNC desktop. Please do not install or remove any python packages after you have activated the shared environment.
*Campaign_driver.py* is the main script that we use to connect to the Carla server and generate data. Run it use:
***"python3 campaign_driver.py --routes_list campaign_configs/routes_userdefined.csv --weathers_list campaign_configs/weathers.csv --output_dir campaign_results/"***
The scenario and weather conditions are stored in the json files in *campaign_configs* folder. However, you do not need to alter any of them for this MP. The results are stored in the folder specified after output_dir. Make sure the output directory is empty before you run the script.

If you have any technical problems regarding running the simulation, please pose your question on Campuswire, and we will try to reply as soon as possible, but please start early and do not wait till the last minute.

**To complete the following tasks, you MUST run the campaign in Task 0 and collect the simulation data.**

Task 1 – Getting to know the analysis environment (6 points)
Before any analysis can be done on a given dataset, you will need to know how to import, handle and do some basic data manipulation programmatically. This task is designed to help you get accustomed to the data analysis environment. Complete the following tasks using Python Jupyter Notebook. **Throughout this MP we recommend using Pandas data frame for data analysis.**

1. Import all the .csv data (including 4 weather conditions:*"clear-night", "clear-sunset", "clear-noon", "rain-noon"*) of the **ghost_cutin** scene into Jupyter

Notebook. List the first 5 rows of the **ctl.csv, cvip.csv**, and **traj.csv** with the **clear-night** weather condition. (1 point)

2. Summarize the following information for each weather condition *("clear-night", "clear-sunset", "clear-noon", "rain-noon")*: (2 point)
   a. The duration of the scene.
   b. Mean and standard deviation of the values of the features (*"throttle", "steer", "brake", "cvip", "x", "y", "v"*). Round your results to 3 decimal place and save them in a table, with the weather conditions as columns, and the features as rows (hint: you can store the table in a data frame). Example format:

| | clear-night | clear-sunset | clear-noon | rain-noon |
|---|---|---|---|---|
| **throttle** | 0.930 (0.212) | 0.086 (0.776) | 0.980 (0.662) | 0.212 (0.746) |
| **steer** | 0.694 (0.516) | 0.783 (0.166) | 0.945 (0.643) | 0.455 (0.608) |
| **brake** | 0.968 (0.385) | 0.521 (0.633) | 0.125 (0.500) | 0.399 (0.295) |
| **cvip** | 0.178 (0.449) | 0.168 (0.291) | 0.948 (0.212) | 0.656 (0.531) |
| **x** | 0.274 (0.409) | 0.414 (0.015) | 0.598 (0.439) | 0.846 (0.292) |
| **y** | 0.520 (0.999) | 0.357 (0.103) | 0.396 (0.641) | 0.569 (0.478) |
| **v** | 0.012 (0.154) | 0.465 (0.862) | 0.545 (0.126) | 0.675 (0.722) |

3. Visualize the campaign results of the **ghost_cutin** scene for each weather condition. Plot the **throttle** values (y-axis) of the agent vs time (x-aixs). Please plot all the weather conditions in one figure, and repeat the same step for all other features as well (*"steer", "brake", "cvip", "x", "y", "v"*). (2 point)

4. Based on your intuition and life experience, which of the features do you think will change during an accident? How will the feature(s) change? By looking at the plots you generated in Task 1.3, combined with your reasoning (without looking at 'route_highway.txt'), which weather condition(s) has an accident? (1 point)

<u>Task 2 – Analysis of simulated data (23 points)</u>

The goal of this task is to let you analyze and identify features in the simulated dataset, in particular abnormal AV behavior caused by adverse driving conditions. Assuming that the AV is well trained under good weather conditions (e.g. ‘clear-noon’), so it behaves normally in these weather conditions.

1. Suppose each simulation run has a result of accident/non-accident, calculate the probability of accident  (counts, marginal probability). Hint: for each run, the collision results are stored in ‘*route_highway.txt’*. You can check the accident status by looking at the ‘status’ field under the ‘record’ section (‘Completed’ means no accident; ‘Failed’ means an accident has occurred). (1 point)
2. By looking at the completion records, and the plots you generated in Task 1, under which weather condition(s) did the accident happen? Does that match your guess in Task 1? When did the accident happen? Why do you think the accident happened at that time? (2 point)
3. Accidents are caused by abnormal AV behavior. However, there are other adverse driving conditions when there are abnormal AV behaviors while no accident occurs. From the plots you generated in Task 1.3, do you observe any other abnormal behavior? If so, what do you think is(are) the cause(s) of this behavior? Complete the following questions to prove your assumption (4 points):
    a. Starting by plotting the distribution of the features for the abnormal runs (including the accident runs) vs normal runs. Treat the values at each time point as an individual sample and generate the density plot of the distribution.
    b. Apply hypothesis testing to support your hypothesis. In this question, you will use 2-sample t-test to test on the **‘steer’** values of **abnormal runs vs normal runs**.; (a) State the null and alternative hypotheses; (b) Perform the test and calculate test statistics; (c) Assume a significance level of 0.05, what is your conclusion?
    c. Does the testing result contradict your observation? Why?
4. Among the features (‘throttle’, ‘brake’, ‘steering’, ‘cvip’, ‘x’, ‘y’, ‘v’), some of them are better indicators of abnormal AV behavior, can you identify them? (3 points)
    a. By looking at the distribution plots of the features in Task 2.3, explain your choice of indicators.
    b. For the fields you identified as good accident indicators above, are they related (Calculate the Pearson correlation coefficient between each pair of the indicators)? If so, how does that affect the predicting power of using one field versus using all of them?
5. Suppose we want to use hypothesis testing to test whether the field you choose in Task2.4 is indeed a good indicator of abnormal AV behavior, using the Kolmogorov–Smirnov two-sample test. (6 points)
    a. Construct the null and the alternative hypothesis and state them below;
    b. Perform the KS two-sample test using the python package;

c. Assume a significance level of 0.05, what is your conclusion?
d. Repeat the same test on an UNSELECTED field in Task 2.4 and repeat the test, what is your conclusion?
e. What are the major differences between the KS test and the t-test?

6. Keeping in mind that this experiment is executed over a period of time, what assumption did you make when using the KS two-sample test on the distributions in Task2.4? Are you able to come up with one situation where the assumption fails? If there is any, please explain. What are the assumptions underlying the test? How does that match your samples (data generation)? (4 point)

7. The dynamic-time-wrapper (DTW) is a method to compare two time-series data (such as the control and the trajectory data collected in our simulation). Use the DTW package in python (dtaidistance · PyPI), and apply the DTW distance on the two time-series dataset: (1) steering data of clear-night and (2) steering data of clear-sunset, using steering data of clear-noon as a reference. What can you say about the DTW distance for (1) and (2) with respect to the reference? (3 points)

## Part 2: Real-world records

**Background:**
Then, we will explore failures and disengagements in Autonomous Vehicles (AVs) being tested on public roads in California. However, while derived from the California Department of Motor Vehicles (DMV) database, the dataset you will be using **has been sufficiently altered to be manageable for this project. As such, the results of the analysis you perform will not directly represent the California study.** The analysis will use statistical and probabilistic approaches to evaluate how well the AI-driven decision and control of AVs work under various conditions and develop insights into why/how they disengage.

The California DMV mandates that all manufacturers testing AVs on public roads file annual reports detailing disengagements and accidents. Disengagement occurs when a failure in the AV system causes control of the vehicle to switch from software to the human driver.

**Dataset description:**

In a real-world setting, the data required for analysis might be spread across multiple fields. Below is a description of the raw file in which the data is available. As mentioned previously, the data has been derived from California DMV, but has been sufficiently modified. Identicality to a known AV manufacturer is purely coincidental.

**mp1_av_disengagement.csv**
This file lists the details of each disengagement that happened in AV testing.

| Column Name | Explanation |
|---|---|
| Month | Month and year when the disengagement happened |
| Car | ID of the AV |
| Location | Where the car was when the disengagement happened |
| Weather | Weather conditions when the disengagement happened |
| TypeOfTrigger | Whether the disengagement was automatic (decision taken by AV) or manual (decision taken by human driver) |

| | |
|---|---|
| ReactionTime | Time taken, in seconds, by the human driver to take control of the car after an automatic trigger.<br>NOTE: ReactionTime is not given for manual disengagements since it does not involve a trigger by the AV. |
| Cause | Reason for the disengagement |

Task 3 – Probabilistic Analysis of AV Disengagement (28 points)

Humans adapt to the lighting and weather conditions while driving. Given that AV technology uses sensors like camera and LiDAR whose performance may vary under different lighting and weather conditions, it becomes paramount to understand if AVs are able to cope with the change in the environment (sensor performance). The dataset provided to you has disengagement measurements under different weather conditions which can help us understand the effect of weathers.

Given below are some assumptions that you will need to do the analysis for this task.

1. There can be at most one disengagement in a mile
2. The total amount of miles driven by all the AVs in the dataset is 505229 miles.
3. A day can be either clear or cloudy, but not both. The probability of a day being clear in California is 72% [5].
4. The AV is equally likely to drive on a cloudy day as on a clear day.

The above assumptions should be enough. However, in case you need to make more assumptions, consult the instructors or make a post on Campuswire. The instructors will respond as quickly as possible.

1. Based on the above assumptions, answer the following questions on basic probability. (6 points)
   a. The assumption of at most one disengagement per mile allows us to treat the occurrence of a disengagement in a mile as a random variable with a __________________ distribution. (1 point)
   b. Based on the above assumptions, calculate the probability of disengagement per mile on a cloudy day. (1 point)
   c. Based on the above assumptions, calculate the probability of disengagement per mile on a clear day. (1 point)
   d. Similarly, calculate the probability of an automatic disengagement per mile on a cloudy day, and the probability of an automatic disengagement per mile on a clear day. (1 point)
   e. How likely is it that, under cloudy conditions, in 10,000 miles, there are 100 or more disengagements? (**hint**: use Central Limit Theorem)(2 points)

2. Assuming that the disengagement per mile is a random variable with the distribution you answered in Task 3.1.a, and the weather condition is *cloudy*. (5 points)

   a. What is the distribution of miles to the next disengagement? Explain your reasoning. Calculate and state the values of the parameters of the distribution. (1 points)
   b. What is the distribution of the number of disengagements in 10,000 miles? (**hint:** this is equivalent to drawing n=10,000 independent trials from the distribution of disengagement per mile you calculated from Task 3.1.a) Calculate and state the values of the parameters of the distribution. (1 points)
   c. Notice that the n in Task 3.2.b is large while the probability p of disengagement per mile is very small, what distribution does the your answer in Task 3.2.b approximate? Calculate and state the values of the parameters of the distribution. (1 points)
   d. Plot the probability mass function (PMF) of the distribution in Task 3.2.b and Task 3.2.c. What do the 2 plots look like? (1 point)
   e. Solve Task 3.1.e by using the cumulative distribution function (CDF) of the distribution you computed in Task 3.2.c and compare the results. Discuss your findings. (1 points)

3. What's the conditional probability that the reaction time is: (Hint, there might be multiple conditions to consider) (4 points)

   a. Greater than 0.4s given that the weather was cloudy? Reaction time is measured only in cases where there was an automatic disengagement. (2 points)
   b. Greater than 0.7s given that the weather was clear? Reaction time is measured only in cases where there was an automatic disengagement. (2 points)

4. A study found that an **automatic AV disengagement** will result in an accident if the human driver is *slow* in reacting. Following reactions are considered slow: (i) a reaction time greater than 0.4s under cloudy conditions and, (ii) a reaction time greater than 0.7s under clear conditions. Find the probability of an accident per mile due to automatic AV disengagement and slow reaction. (2 points)

5. Next, you will investigate how to diagnose the cause of an AV disengagement based on new observations. (3 points)

   a. An AV had a disengagement with a reaction time greater than 0.4s on a cloudy day. What is the posterior probability that the root cause of the disengagement was "Software Froze"? (2 points)
   b. How do your conclusions in Task 3.4.a change if the disengagement happened on a clear day with reaction time greater than 0.7s? (1 point)

6. In this question, you will construct a Naive Bayes model to infer the root cause of disengagement scenarios of AVs. Naive Bayes assumes that the factors are class conditionally independent. We assume that both Location (urban-street or highway) and Weather (cloudy or clear) are factors related to the Cause (consider the Cause has 3 different values, “Software Froze”, “Hardware Fault” or “Other”), and Location and Weather are independent given the Cause. Infer: What is the Cause of the disengagement given that we can observe the Location and Weather? (8 points)

   a. Draw a graph for the Naive Bayes model described in the question. (2 points)
   b. Count the number of parameters needed to define the Naive Bayes model (including the prior and the conditional probability distributions) .(2 points)
   c. Based on the number of parameters needed, derive and show the conditional probability tables and prior probability from the given dataset in order to infer the Cause. (2 points)
   d. According to the conditional probability tables you derived, what is the most possible root cause given the day was cloudy and the Location was urban-street. (2 points)

## Part 3: Combine Analysis of Simulation Data and Real Data

**Background**:
As the final part of this MP we will explore the connections, the similarities, and the differences between the California DMV real-world dataset and a baseline simulation dataset. In particular, we would like to understand the relationship between the baseline simulation dataset and the AV's real-world behavior, and whether the simulated dataset can sufficiently represent real-world autonomous driving. The baseline simulation dataset consists of the results of 500 runs of Carla simulation(the results are intentionally altered to fit the problem) and it contains accident information under different weather conditions: (i) clear, (ii) cloudy, (iii) rainy, (iv) snowy. We will provide you with this dataset on the course website.

<u>Task 4 – (10 points)</u>

1. Parse the provided Carla simulation dataset and calculate the following probabilities for the cut-in scenario (you need to filter our invalid data points before doing the analysis):
    a. The probability of accident P(acc=1) across all weather conditions.(1 point)
    b. The probability of accident conditioned on the weather, P(acc=1 | weather=?), for each weather condition.(1 point)

2. The baseline simulated dataset contains the accident information under snowy conditions and under rainy conditions. In California it is sunny 72% of the time, rainy 10% of the time, snowy 3% of the time and the rest of the time it is cloudy. In Chicago, it is sunny 56% of the time, it rains 25% and it is snowy 9% of the time, and the rest of the time it is cloudy.
   Can you use the baseline data to project the probability of accident in the cut-in scenario to California and Chicacgo, respectively? Clearly state your assumption and your method.(2 points)

3. In Part 2, Task 3.4, you calculated the AV's probability of accident per mile for the AV for California DMV dataset. Suppose you want to compare the simulated accident rate with the real dataset accident rate.(6 points)

   *Hint: For this question, you can assume that the probability you calculated in Part 2, Task 3.4 is the AV's marginal (unconditional) accident rate per mile for the real CA DMV dataset, and you can further assume that in simulation, there is one cut-in scenario (run) per one mile.*
    a. Unfortunately the real DMV only has sunny and cloudy weather, how would you make a reasonable comparison between the probability of an accident of the simulated dataset and the real dataset in this case? What do you observe? (2 points)
    b. Suppose that there are **k** cut-in scenarios per one mile, and the probability of having an accident in one cut-in scenario is ***p.*** Starting with the Bernoulli trial, what is the probability to have at least one accident per one mile? State your reasoning in detail.(2 points)

c. Can you provide some explanations on the observation? Why do you think the probabilities are so different?(1 point)
d. What are the pros and cons of simulation given the discrepancy between probability of accident in simulation and the real-world data? (1 point)

4. Given that this is the first time that the course offers an MP that involves complex simulation together with comparison with real data, what are your suggestions to improve the MP? What difficulties did you encounter? Please list your suggestions for improvement.