# Project Summary

Due to all the restrictions put in place during the COVID-19 pandemic, it has become increasingly more difficult to find restaurants. Our model will determine which restaurant accommodates the most preferences of the user, such as price range, dietary restrictions, and delivery or take out options. Our model will take into account a wide range of personal preferences of the user, and find the best fit restaurant for them to eat at.

Propositions

**low**: Represents the lowest price range, true when the restaurant can provide a meal within the range

**med**: Represents the middle price range

**high**: Represents the highest price range

**restriction\_gluten**: True if the restaurant can provide gluten free meals

**restriction\_vegan**: True if the restaurant can provide vegan meals

**restriction\_lactose:** True when the restaurant can provide lactose-free meals

**restriction\_vegetarian:** True when the restaurant can provide lactose-free meals

**dine\_in**: True when the restaurant allows the user to dine-in

**take\_out**: True when the restaurant has a take-out option

**delivery**: True when the restaurant has a delivery option

**under\_10**: Under a 10 minute walk from Queen’s campus

**10\_to\_20**: Between a 10 to 20 minute walk from Queen’s campus

**over\_20**: Over a 20 minute walk from Queen’s campus

# Constraints

**Price**

* + Lowest price point:
    - (low ^ ~med ^ ~high)
  + Medium price point:
    - (med ^ ~high)
  + High price point:
    - (high)

**Dietary restrictions**

* + (restriction\_gluten **∧** restriction\_vegan **∧** restriction\_lactose **∧** restriction\_vegetarian) - the restaurant can accommodate all four types of dietary restrictions
  + (restriction gluten ∨restriction\_vegan ∨restriction\_lactose ∨ restriction\_vegetarian) - the restaurant can accommodate at least one type of dietary restrictions
  + restriction gluten **∧ (** restriction\_vegan ∨restriction\_lactose ∨ restriction\_vegetarian) - the restaurant can accommodate two types of dietary restrictions
  + restriction gluten **∧** restriction\_vegan **∧ (**restriction\_lactose ∨ restriction\_vegetarian) - the restaurant can accommodate three types of dietary restrictions

**Methods of consumption**

* + Dine in:
    - (dine\_in ^ ~take\_out ^ ~delivery)
  + Take\_out:
    - (~dine\_in ^ take\_out ^ ~delivery)
  + Delivery:
    - (~dine\_in ^ ~take\_out ^ delivery)

**Distance**

* + Under 10:
    - (under\_10 **∧** ~10\_to\_20 **∧ ~**over\_20)
  + 10 to 20:
    - (~under\_10 **∧** 10\_to\_20 **∧ ~**over\_20)
  + Over 20:
    - (~under\_10 **∧** ~10\_to\_20 **∧** over\_20)

# Model Exploration

We have explored our model in the following ways throughout the course of our project:

**How we modified / improved our propositions and constraints:**

* Initially, three of our propositions for distance were under\_10, under\_20, and over\_20. One of our constraints was that a restaurant must be under a 10 minute walk, under 20 minutes, or over 20 minutes. In order for us to categorize and offer the top pick between the restaurants, the restaurant must be one, and only one, of the propositions. The constraint format we chose to use was ~(under\_10 ∧ under\_20), ~(under\_10 ∧ under\_20 ∧ over\_20), and etc. After analyzing these constraints, we realized that the propositions should be under\_10, 10\_to\_20, and over\_20 instead. The constraint ~(under\_10 ∧ under\_20) posed a problem because a restaurant that was under a 10 minute walk away would also be a 20 minute walk away. With this constraint, this restaurant would not be considered for the top pick since it would return False.
* When starting our project, we had made a proposition for opening time and closing time of the restaurant. However, after careful consideration we decided to take these propositions out since it did not make sense to model whether the time a user wanted to go to a restaurant was within the opening and closing time of the restaurant with logical propositions. It would not have made sense, or would have over complicated the problem, to assign a truth value to a proposition modelling this.
* We also started out with a proposition for the type of cuisine a restaurant offers. This proposition was taken out as well since in order to model it as a logical proposition, we would need a true or false value for each type of cuisine. After exploring the scope of our project, we realized that it would be unnecessary to have each type of a cuisine as a proposition, since other requirements would be of greater importance to the user.
* Our initial plan to represent the price range of a restaurant was to use the proposition mi. The proposition was true when m was less than or equal to i. We realized this was not the best way to represent this criteria. Instead, we replaced it with price\_$, price\_$$, and price\_$$$ to represent three ranges of prices. Each restaurant would be True for one of the propositions and False for the others for clarity.

**How we modified / improved our code:**

* In our code, we started by storing our restaurants in a 2D list. After hearing feedback from peer assessments, we decided that the restaurants could be better modelled if they were stored as a list of restaurant objects. This is because it is easier to store information on each restaurant using an object. This method is more time efficient since there would be less iterating through lists and it is easier to see how everything is stored. As well, we added a csv reader function as a way to read in the restaurant objects.
* Our first approach to our modelling was to add in all of the constraints to check if they satisfied the model. However with this approach, we didn’t have a way of saying that a restaurant satisfied the model or ranking/sorting the restaurants. We changed our approach to the problem and split the constraints up, making a separate function to evaluate each constraint. This means the function can give us a numerical value for each constraint allowing the restaurants to be compared and sorted into a ranked list.

First-Order Extension

We could extend our model to a predicate logic setting by changing the propositions and constraints to the following:

* A(x): Represents the lowest price range, true when the restaurant can provide a meal within the range
* B(x): Represents the middle price range
* C(x): Represents the highest price range
* G(x): restaurant can provide gluten free meals
* V(x) True if the restaurant can provide vegan meals
* L(x): True when the restaurant can provide lactose-free meals
* P(x): True when the restaurant can provide lactose-free meals
* D(x): True when the restaurant allows the user to dine-in
* T(x): True when the restaurant has a take-out option
* E(x): True when the restaurant has a delivery option
* W(x): Under a 10 minute walk from Queen’s campus
* Y(x): Between a 10 to 20 minute walk from Queen’s campus
* Z(x): Over a 20 minute walk from Queen’s campus

**Extended Price**

* + ∃x(A(x) ∨ B(x) ∨ C(x)) - There exists a restaurant that is in one of the price ranges
  + ∀x(~(A(x) **∧** B(x)) - All restaurants cannot have more than one price range

**Extended Dietary restrictions**

* + ∃x(G(x) **∧** V(x) **∧** L(x) **∧** P(x)) - There exists a restaurant that can accommodate all four types of dietary restrictions
  + ∃x(G(x) ∨V(x) ∨L(x) ∨ P(x)) -There exists a restaurant that can accommodate at least one type of dietary restrictions
  + ∃x(G(x) **∧ (** V(x) ∨L(x) ∨ P(x))) -There exists a restaurant that can accommodate at least two types of dietary restrictions
  + ∃x(G(x) **∧** V(x) **∧ (**L(x) ∨ P(x)) -There exists a restaurant that can accommodate at least three types of dietary restrictions

**Extended Methods of consumption**

* + ∃x(D(x) **∧** T(x) **∧** E(x)) - There exists a restaurant that offers all three methods of consumption
  + ∃x(D(x) ∨ T(x) ∨ E(x)) -There exists a restaurant that offers at least one method of consumption
  + ∃x(D(x) **∧** (T(x) ∨ E(x)) - There exists a restaurant that offers dine-in and one other method of consumption

**Extended Distance**

* + ∃x(W(x) ∨ Y(x) ∨ Z(x)) - there exists a restaurant that is at least one distance
  + ∀x(~(W(x) **∧** Y(x) **∧** Z(x))) - all restaurants must not be three of the options