POL 495 Summary Report of Code + Analysis

Background

Over the course of the semester I learned a lot about social feedback affecting declining public infrastructure quality and how it can create an impact in the community over a long term. This was the main question that was explored in the behavioral experiment by Dr. Yu. The main premise of the experiment was the dynamic between the user (farmer) and provider.

The main question of my research was asking how the farmer-provider interaction surrounding public infrastructure of a water irrigation system would be affected when the provider or the farmer is controlled by an artificial intelligent algorithm instead of a human subject. Developing a conceptual model surrounding this use case was extremely helpful in implementing my code and development for the project. Figuring out the development was important and I did a lot of research surrounding this. A lot of experimental economics and treatments were designed where researchers substituted human subjects for automated or CP subjects instead. This was to eliminate social preferences or induce different types of behavior from the human subjects in the experiment. Understanding how the CP worked with these experimental designs helped me theorize a conceptual plan for automating the experiment. I discuss my findings and theoretical implementation in Assignment 2.

The main strategies that I decided was needed in the experiment were:

1. Tit for Tat
2. Altruistic
3. Nash Equilibrium

Tit for tat would involve the decision to be made by the provider when the CP is the provider, and also when the CP is a farmer. Altruistic and Nash equilibrium were focussed on the Provider's decisions when the CP is a provider.

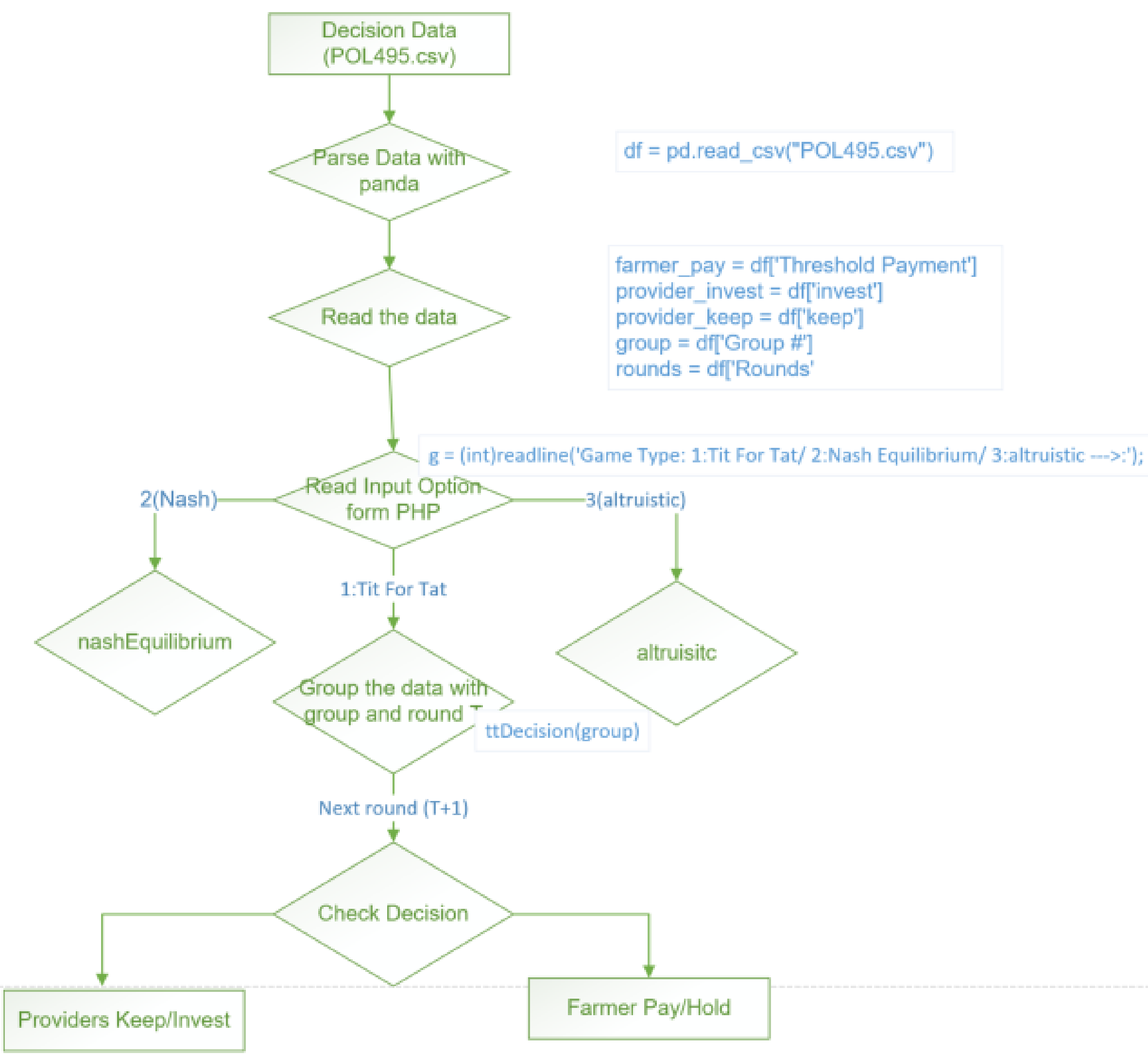
Analysis of Code

Example in Experiment of Round 4 showing how Farmers Pay or Hold and Providers Invest or Keep the money. This results in the irrigation system being either good quality or poor quality.

RESULT

	Decision	Tokens earned ($ earned)
Farmer 1	PAY	15 ($0.75)
Farmer 2 **(You)**	PAY	15 ($0.75)
Farmer 3	HOLD	10 ($0.5)
Provider	INVEST	5 ($0.25)

I have outlined the main functions and the steps taken by the program to follow this using an Entity-Relationship Diagram. The ERD Diagram shows how the decision data is processed, read, and the functions are created to showcase the decisions made by the Farmer or Provider.



I performed logistic regression as my data analysis process to understand the accuracies and help with making predictions in the data. This aspect helped with making predictions based on randomized decisions by both user and provider. Using yhat and np.random tool, I showed how user provider input is randomized and the infrastructure state outcome is predicted accurately.

Random Value Predictions

Make predictions based on randomized decisions by both users and provider

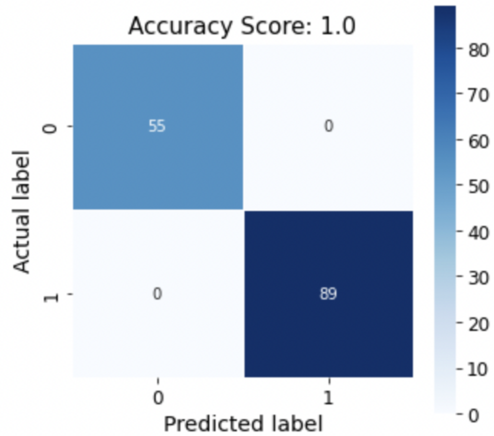
In [179]:
```
random = np.random.choice([0, 1], size=(20,3))
random
```

Out[179]:
```
array([[1, 0, 1],
       [1, 0, 1],
       [1, 0, 0],
       [0, 1, 1],
       [1, 0, 0],
       [1, 1, 0],
       [0, 0, 0],
       [1, 0, 1],
       [1, 1, 0],
       [0, 0, 0],
       [1, 0, 0],
       [1, 0, 1],
       [1, 1, 1],
       [1, 1, 1],
       [1, 0, 0],
       [0, 0, 1],
       [0, 1, 0],
       [0, 0, 1],
       [0, 1, 1],
       [0, 0, 0]])
```

In [180]:
```
# make predictions on the entire training dataset
yhat = model.predict(random)
# connect predictions with outputs
for i in range(10):
    print(random[i], yhat[i])
```

```
[1 0 1] 0
[1 0 1] 0
[1 0 0] 1
[0 1 1] 0
[1 0 0] 1
[1 1 0] 1
[0 0 0] 0
[1 0 1] 0
[1 1 0] 1
[0 0 0] 0
```

Next, I looked at performing predictive analyses to better understand the data with Confusion Matrix and Decision Trees. The confusion matrix shows that there are high True Positive and True Negative values. This is really great in understanding our data because True Positive (upper left) is an outcome where the model correctly predicts the positive class. True Negative (lower right) is an outcome where the model correctly predicts the negative class.



Following the sequence of instructions and the decisions made by either the Provider or the Farmer, the code functions take into account the automation of decisions of both parties. Throughout the code, I used the column Threshold Payment instead of individual farmers in order to predict the decision of the third farmer aka the CP. I used the Threshold Payment to understand the decision made by two farmers in order to predict the third farmer. Each individual round's analysis is done when T is used and the next round is shown as T+1.

First, the code performs operations that the Provider performs.
The first is the Nash Equilibrium behavior.
The **nashEquilibrium()** function looks through the Threshold Payment column of the dataset and maintains payment of the provider to be 0. This function makes the provider corrupt and does not allow the Provider to maintain the irrigation system regardless of whether the farmers pay or not. This correlates with the function of round T.

The second is the Altruistic behavior.
The **altruistic()** function performs the opposite behavior of the NashEquilibrium() function. The provider invests money regardless of the behavior of the farmers. This shows that the provider is investing money in an altruistic way.

The third is the Tit for Tat Behavior. If the Threshold Payment (farmers pay) is 1, which means they paid, the provider will also invest money in the irrigation plant. This is represented and shown in the function as farmer_pay. However, if the farmer_pay = 0, the provider will also not invest and we append 0 to the result1 array which shows that the provider did not invest. This is shown by the function **ttProvider()**

There is also a Tit-for-Tat function that is represented which involves the previous rounds, and shows adaptive behavior. This function creates the knowledge of understanding how the provider works when knowing the infrastructure state. If the provider chose to invest in Round T and infrastructure State was 0 in Round T, the provider responds by NOT investing in Round T+1. Next, if the provider chose to invest in Round T and infrastructure State was 1 in Round T, the provider responds by investing in Round T+1.

To start the next round's function, we need preliminary functions to help guide the decisions of farmers or providers in round T in order to establish their decisions for round T+1.
We have **farmer_decison(T)** and **provider_decison(T)** to help establish basic logic which will be called in the functions ttDecisionT1(group, r1) and ttAll(). This logic shows the decisions made by the farmer and provider in round T.

The function **ttDecision(group)** helps with understanding how farmers and providers decisions are played out in regards to groups. Each group correlates to the group made with human subjects acting as the farmers and provider. This function helps the user control and view decisions based on Group 1, 2, 3, etc. Since the code is utilizing the dataset to model the predictions, each group can be separated in order to analyze behavior per group, or combined and looked as a whole where all the groups are together.

The combination of the tit-for-tat behavior of the farmers' decisions and the provider's decision creates the final function. This function has the objective of implementing farmers' decisions for the next round, as well as the provider's decision for the next round. These are the functions **ttDecisionT1(group, r1)** and **ttAll()**. These functions help analyze how everything works in conjugation with each other. This involves the logic of next rounds as well as groups to create a function that has predictive functioning of both the farmer and the provider's decisions. The main logic of rounds is as follows:
If the provider chose to invest in Round T and infrastructure State was 0 in Round T, Provider responds by NOT investing in Round T+1.
If the provider chose to invest in Round T and infrastructure State was 1 in Round T, Provider responds by investing in Round T+1.
If the farmers chose to pay or the Threshold Payment is 1 in Round T and infrastructure state is 0, farmer holds (does not pay) in Round T+1.
If the farmers chose to pay or the Threshold Payment is 0 in Round T and infrastructure state is 1, farmer pays in Round T+1.
Both the farmer and the provider Tit-For-Tat decision-making is happening simultaneously.

The **main()** function runs the code. When executed, if the user plays 3, the provider is altruistic in their choices. If the user plays 2, the provider executes Nash Equilibrium qualities. And if the user plays 8, the function shows Tit for Tat involving rounds for both decisions of farmer and provider.

This means, both the farmer and the provider Tit-For-Tat decision-making is happening simultaneously. This logic is implemented by the final function ttDecisionT1(group, r1). This is called by the main() function and the automation is complete.

Video Recording of Main Function:
https://www.loom.com/share/cbca28671b034392b609de8794bb252e