Infection prediction.

In this problem, we are interested in predicting the spread of a disease between vaccinated and unvaccinated individuals. In order to solve this problem, we have collected data for the past 300 days. The data contains the number of unvaccinated infected individuals, number of vaccinated infected individuals, and total number of vaccinations. The goal is to create an algorithm that would predict the total number of infections during the following 100 days.

Data provided: You have access to three csv files. Each file contains the artificially generated data for 3 populations. You can find the total population size in the following table.

| Data file | Total Population |
| --- | --- |
| observations\_1.csv | 25’000.000 |
| observations\_2.csv | 6’000.000 |
| observations\_3.csv | 500.000 |

Task 1 [30 points]. Give a prediction for the following 100 days for each one of the three populations provided in the data files. The performance on each data set will be evaluated using the mean square error to the real infection numbers for each population. You will provide those predictions as a csv file with the following format:

Predicted\_infections

13,

23,

25

In this example, 13 corresponds to the prediction on the first day, 23 corresponds to the prediction on the second day, and so forth and so on.

Task 2 [30 points]. Produce a python library with a function with the following function:

model\_prediction(data\_csv, number\_of\_days)

This function should save a csv file with the name “predictions.csv” which contains the prediction for the next number\_of\_days for a model trained on the data file data\_csv. The predictions should be given as a csv file that contains the predicted number of infections.

Your model will be trained on 3 different data sets, and the performance will be evaluated by the mean squared error in the following 100 days.

Task 3 [20 points]. The quality of your code will be taken into consideration (code documentation, and clarity)

Task 4 [20 points]. Document your methods and algorithms. Produce a readme.md file that explains your solution. In your readme file, include instructions to install your library (including python version, and dependencies), how to use your library, and the algorithms underlying your solution. The readme.md file will account for 20 points.