**EECS6895 Milestone 1 Report**

**COVID-19 Risk Prediction via Bayesian Network**

**Topic:** A11: Reasoning (Understanding Causalities via Bayesian Network and/or Others)

**Team:** Yiwen Fang (yf2560) | Guoshiwen Han (gh2567)

**Contents**

**1. Task Goal**

**2. Literature Survey**

2.1 Previous Work

2.2 Software and Tools

**3. Methodology**

**4. System**

**5. Timeline**

5.1 Milestone 1

5.2 Milestone 2

5.3 Milestone 3

5.4 Final project

**6. References**

**1. Task Goal**

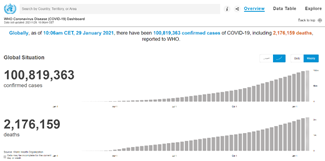
The primary achievement of the project is finishing a three-step process. Step 1 is a cycle of implementing training data, training Bayesian models, doing testing and model validation. Step 2 is getting user input to make a prediction. Step 3 is generating the probability of risk and outputting the result to the user. Naive Bayes is the underlying key algorithm. A visualized panel is designed as a UI for the application.



**Figure 1.** UI panel [1]

There are three reasons making the research and development important, which are user-friendly interactions, reasonable predictions, and adaptive models. Interactions are interactions between users and the web application, e.g., receiving user input and generating output back to users. Predictions are the predictions or forecasting of future status. Adaptive models allow themselves to dynamically update models in a proper frequency, e.g., every week, since the virus variation and the sample data is kept updated. This automatic updating life cycle is similar to the three-step process consisting of training data, training Bayesian models, testing, and model validation.

However, we are also facing big data challenges, i.e., volume, velocity, and variety. The volume of COVID-19 data is dramatically expanding and increasing day by day. Our database system can experience pressure on storing and manipulating large volume data. The second challenge is the long processing time for data import, data cleaning, and model training since large volume slows down the velocity. The last challenge is the various types of data in our project, including but not limited to geographical coordinates, time, amount, statistical ratio, and resources.



**Figure 2.** WHO website [2]

**2. Literature Survey**

**2.1 Previous Work**

The literature survey is characterized into three tasks of collecting data, confirming the conjecture based on serological testing, and calculating probability. WHO has done an excellent job of organizing the COVID-19 data all over the world and drawing out the real-time trend line. Martin Neil, et at. worked on the Bayesian network analysis to confirm the conjecture of the bias of reported data based on the Covid-19 serological testing data. Georgina Prodhan and Norman Fenton presented an extension of an existing Bayesian network model for an application in which people could add their own personal risk factors to calculate their probability.

|  |  |
| --- | --- |
| https://lh6.googleusercontent.com/0O7PjsKSX0rU_VvX-33MEj_LCrbkHKP2agMe8Th__3yEN27tbrDxV5QFZ946ZfJ0iliMxvAws75Wha_seK3lMDyeh5DvcVLAoiH9hvtlV3qB525FVPkjZJG4MdmVqw073S6W46oq | https://lh3.googleusercontent.com/js7LEv6gzviLl8Ko9LF-Y2UdKKhGSI6YeYNslD8znQkWeCgNaZdAEBIelZJhmCuhdwtWImidEwMpBBP42KvDMmzpfGxY3vfarTKab38Mim04ktVGiF-fiuLkc5xyVLLf35bVtq52 |

**Figure 3.** WHO real-time data [2]

|  |  |
| --- | --- |
| https://lh4.googleusercontent.com/lqyNO2fw1M7F53ACjnfWr_T5DaHjXAJzbV-hoxbZkBYt643RsgWmtDtdJO8pGakV-LklRZq-CKZfpUOJvb0xal43Dh4tiH3vdjq6jRLcfSsUO6sE6VlQO2IxrmjcXbbbOsMZUz40 | https://lh6.googleusercontent.com/e6YFWNYuqk6s_JM60X36ryq_tD7-I-Z--uzOYrXFI9uWo3Qigca61vFwSo8qVEB_UeyNsu2rWwKgC-h_P6MmpwHkAUIOcEY3ClOaKr-t2O7Xwq_HoIvzc2uMcHE3YBf-etZrjacd |

**Figure 4.** Bayesian network analysis of Covid-19 data reveals higher infection prevalence rates and lower fatality rates than widely reported [3]

|  |  |
| --- | --- |
| https://lh4.googleusercontent.com/6-c2iZRL7ok1Gvf9_HN_YkZSJKX4pxxlQRm6M_oBvkzvPGOMLxVrpxYY2TAFgyHFisBBrLGRiUfN4mU1GlEqdSD38Bdkc2PtZXyIoswPJLpA15HLx4Pyt5LAN7eucPzjiQSVnGzf | https://lh3.googleusercontent.com/NISQGN1rP7MBxkkxYfp1p1KsMOPzE30oh4rf6vvrTqdn4hE52ksT7jlA1BtkORKJ2frMJWfb5olNHdS5s9EFkB6MMphXKo4kOK8Bt58RMZdj6Pez1sWFMIHq3rvVJ_jUq_8KHdh0 |

**Figure 5.** Extending the range of COVID-19 risk factors in a Bayesian network model for personalised risk assessment [4]

**2.2 Software and Tools**

Unlike the statistical software used in the papers such as AGENARISK, Python is used as the underlying language for modelling to integrate with other functions. We split our application into five components.

Models are trained by Python and TensorFlow. Python and its packages are powerful for Bayesian model training. Unlike the statistical software used in the previous papers, models trained from Python are easy to be integrated with code of the web application.

The web application is developed by Python and Django. Django is a web framework integrated with frontend and backend, which is more powerful than Flask.

The database system is based on NoSQL and Spark. NoSQL is a better database choice than SQL due to the flexibility and speed. Spark is a candidate for big data databases.

The server is hosted by AWS, Kafka, and Apache. Deploying to the cloud ensures remote access, distribution, and scalability.

The data visualization is done via Matplotlib, Flourish, and D3. Matplotlib is easy to integrate with Python. Flourish is a commercial software best for dynamic running bar charts. D3 is easy to integrate with JavaScript.

**3. Methodology**

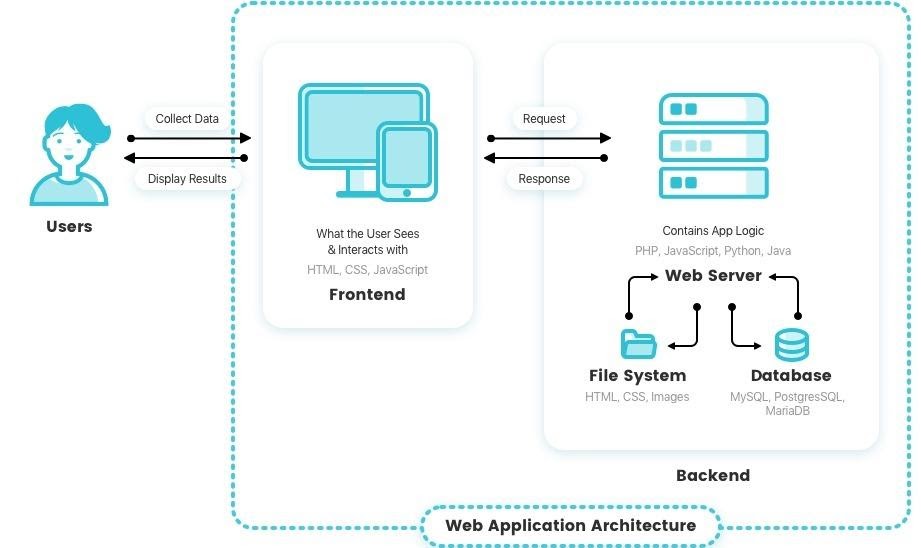
We plan to imply Bayesian network algorithm and PCA (principal components analysis). The datasets we preferred are WHO, Johns Hopkins University, and CDC datasets, and we may make some adjustment according to our later work and need.

|  |  |
| --- | --- |
| https://lh4.googleusercontent.com/CdpbeJJFUZqEfvOrIQh8bOHUd2wObffBhUL55ycl828VUcWj71NDYWywTKqB3-E4pU_16h8o3PO4LGizPZ90UqoR8kcac9bAln989LD-mgrJYQf72i2N2ii9QEVw6aDIjETScVRA | PCA - Principal Component Analysis and Visualisation of Training Load Data |

**Figure 6.** Bayesian Network algorithm (left) [5] and PCA (right) [6]

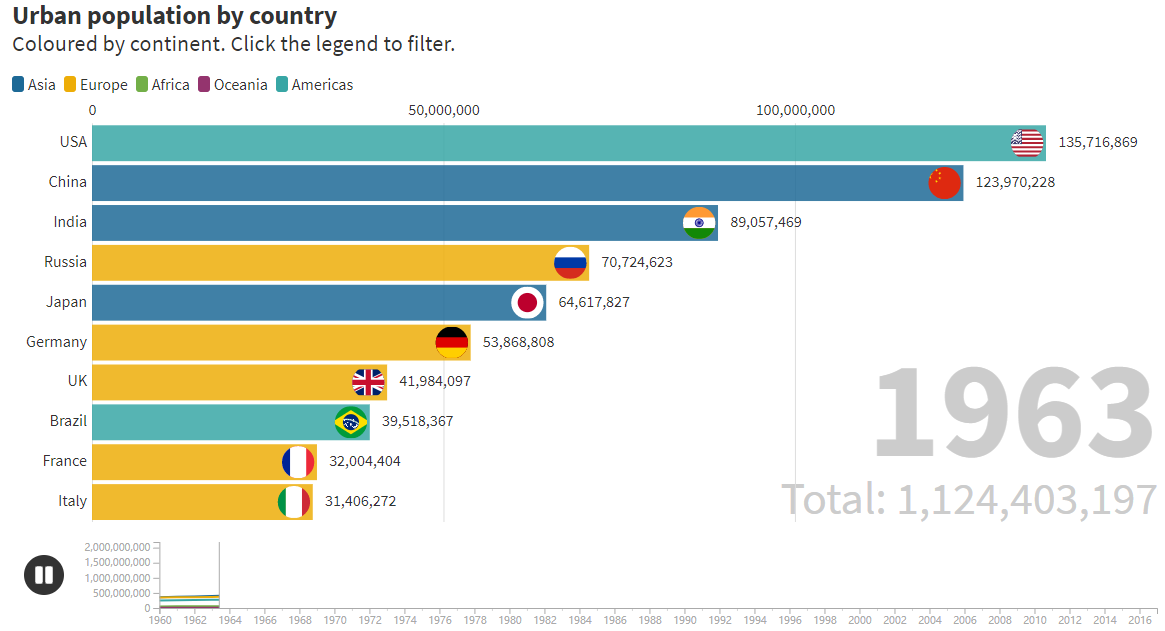
**4. System**

The final system is a web application integrated with frontend, backend, and model. Model is presented via a web application; Frontend accepts user input and transfers it to the backend via API; Backend receives the input data and the model processes the input parameters. After the model finishes processing, the backend outputs the result to the fronted end to the user.



**Figure 7.** System architecture schema [7]

Visualization is realized via tables, static graphs, dynamic graphs. UI enables the users to zoom in for the detailed or zoom out for the scope of results. It interacts with the user by accepting user input.



**Figure 8.** Dynamic running bar chart demo [8]

**5. Timeline**

**5.1 Milestone 1**

Has been Completed

**5.2 Milestone 2**

(1) Data collection and procession

(2) Determine the training method of the model

(3) Coding work

**5.3 Milestone 3**

(1) Model training and evaluation

(2) Frontend, backend, API, database

**5.4 Final project**

(1) Deployment and testing

(2) Extend the result to other factors such as social indicators

**6. References**

[1] https://www.fiverr.com/sulehrinauman/design-creative-dashboard-admin-panel-ui-with-fully-responsive

[2] https://covid19.who.int/

[3] https://www.tandfonline.com/doi/full/10.1080/13669877.2020.1778771/

[4] https://www.medrxiv.org/content/10.1101/2020.10.20.20215814v1

[5] https://www.google.com/url?sa=i&url=https%3A%2F%2Fmedium.com%2F%40pranav3nov%2Fnaive-bayes-for-machine-learning-b81ddf954031&psig=AOvVaw027xnh4qG3gUV8oPOQbnfe&ust=1612607144075000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCNCR3fjD0u4CFQAAAAAdAAAAABAK

[6] https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.sportscidata.com%2Fperform-principal-component-analysis-pca-of-training-load-data%2F&psig=AOvVaw3mh1RDSywwA9e9H6fviYC\_&ust=1612607063109000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPjp1c\_D0u4CFQAAAAAdAAAAABAD

[7] https://reinvently.com/blog/fundamentals-web-application-architecture/

[8] https://flourish.studio/examples/