Capstone Project Final Report:

Injury Forecasting Application Implementation to Maximize Workers' Safety

Client: PA Department of Labor & Industry

CMU Students: Nathan Nayda, Yoshinori Ushiyama, Spencer Huang, Qingyang Shen, Xinye Li

Advisor: Brian J Kolowitz

Table of Contents

Executive Summary

CMU Heinz students cooperate with the PA Department of Labor & Industry to develop a web application on work injuries and illness prediction. This is the eighth version of the whole project and would lay focus on both deployment usability and functions improvements. The CMU students understood that the Pennsylvania worker`s compensation program was established to support medical treatment and compensation for lost wages associated with work-related injuries and illness. The vision towards a final product is to have an efficient prediction tool covering various industries across different counties in Pennsylvania. To achieve this goal, this CMU team has provided solutions on better deployment processes, enhancements on GUI features and logging functions, verification on previous prediction code. 2 different previous prediction models were both added into the web application. This improved web application is more user-friendly as well as developer-friendly, and has been successfully deployed in PA staff`s environment. Therefore, PA Department of Labor & Industry is able to utilize prediction results for education and training and outreach to better prevent injury and illness incidents.

Project Objective

The vision toward a final product is to have a web application that presents the workplace injury and illness prediction results in PA, covering various industries and different counties.

The previous CMU teams have developed prediction web applications, but the application has not been used in the actual business environment and following issues remain challenges.

- Non-compliant software
- Production environment-compatibility
- Lack of testing and existing bugs
- Complex usage of Framework

The CMU team is determined to provide the following solutions to solve problems and improve user experience.

- Software porting and simplification of framework usage to fit production environment
- Enhance GUI features:
  1. Selections on prediction models and granularity
  2. Prediction progress bars
- Bug-fix and security/style enhancement
- Code verification
- Deployment process improvement

Business Impact

Usability: CMU team has successfully deployed web applications into PA Department of Labor & Industry environment, this tool can now be put into full use.
User-friendliness: 2 previous models can be chosen to apply on predictions. A prediction progress bar is added so that PA staff would know how much progress has been accomplished in prediction.
Decision making: Through Power BI dashboard, the prediction results are fully demonstrated for PA staff to make better decisions towards the goal of 1% per year work-related injuries decrease, which is 1,700 cases reduced and $28 million dollar saved.

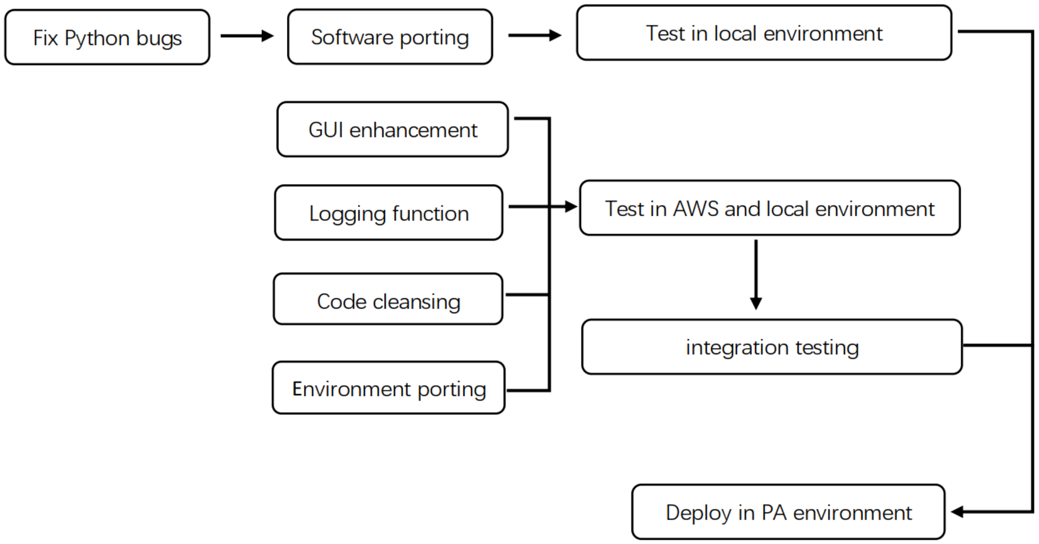
All these improvements would empower the clients to better allocate resources and enhance their training systems.

Project Methodology – Approach

The CMU team applies the following approach to address the challenges and objectives. Before the mid-term stage, fixing bugs and finishing the software porting is our priority so as to have a usable web application. After we tested in AWS and the local environment to make sure the application is running properly, we moved on to several functions that would be beneficial to users as well as future developers.

The CMU team finished all the developing work in time, after testing in AWS and local environments, the project was delivered to PA Department of Labor & Industry and was successfully deployed.

[Fig 1] Approach Flow Chart



[Fig 2] Project Scope

Scope Area	Proposed Action Item	Priority
Fix Python bugs	Fix identified issue ‘nanmean’	1
Port software from “Linux” to “Windows”-based version	Replace Docker with another solution	2
	Move flask functionality to Django	
	Improve build process (MS SQL implement)	
Enhance GUI	- Add functionality to select different prediction models - Add ability to select subsets of counties and codes	3
	- Progress bar for prediction process	
Security and Code Quality Control	Logging and code cleansing (for future development)	4

Project Methodology – Individual Methodology

Objective of each methodology

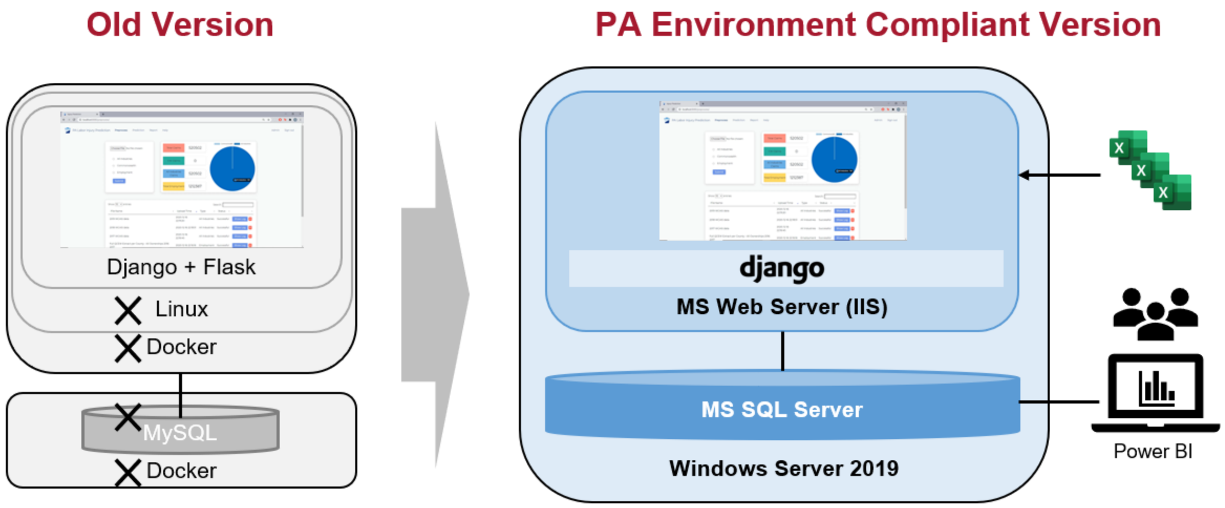
- **Software Porting/Deployment improvement + Compliant to the Production Environment:** In order to operate the system in the client's internal environment, remake it into a system that conforms to the customer environment.
- **GUI Enhancement:** Improve the convenience of end users for use in actual business.
- **Security Feature + Code Cleansing:** In order to operate the system in the production environment, strengthen security functions and check code to improve quality.

Software Porting/Deployment improvement + Compliant to the Production Environment

Since the internal environment of the client is mainly Microsoft products, it is necessary to recreate the application based on Microsoft products in consideration of future support. In addition, it became necessary to avoid using Docker due to the charge for Docker licenses. In the old version, multiple frameworks were used for one application, and it was necessary to use the frameworks simply in consideration of future development efficiency improvement.

The CMU team decided to replace the old version of the application, which was based on Linux + Docker + MySQL, with a Windows + MSSQL application that is compliant with the client environment, and to make the best use of the Django framework and remove the Flask framework. We also introduced an IIS Web server to provide the application as a Web service in the client environment.

[Fig 3] New Architecture



GUI Enhancement

The Graphical User Interface (GUI) enhancements were mainly based on functionality needs and improving user experiences. As described by the client, the application contained a few unnecessary outputs and there was also a missing functionality that would allow the user to enter in extra information to enhance the prediction results. Considering the collaborative nature of the project, the CMU team followed the following process when creating a new GUI enhancement:

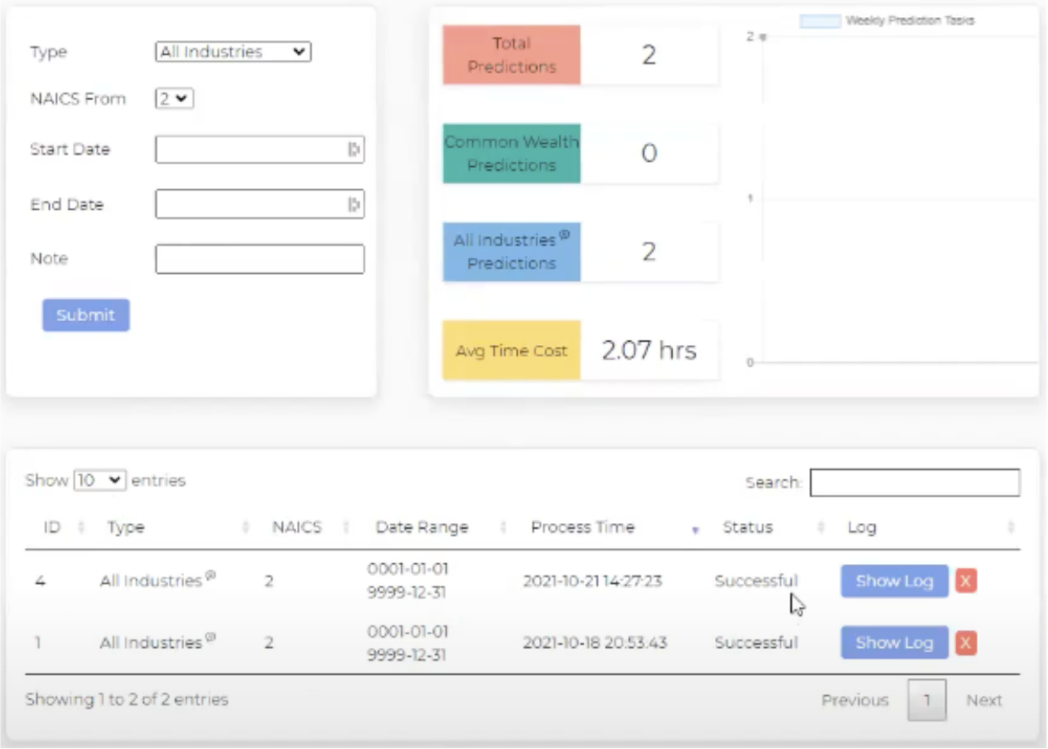
1. Create mockup of GUI enhancement
2. Review mockup with client during weekly meetings
3. Implement GUI enhancement on the application
4. Confirm outputs with client

The CMU team implemented the following GUI enhancements:

- Progress Bar for uploading a dataset and running prediction model
- Remove statistic chart from predictions page
- Create new statistic page with new and improved statistics
- Add functionality to select different models, counties, NAICS Levels, and NAICS Codes
    - Changed UI in predictions page to allow for this functionality

[Fig 4] Original Predictions Page

Old Predictions Page



Type: All Industries

NAICS From: 2

Start Date:

End Date:

Note:

Total Predictions: 2

Common Wealth Predictions: 0

All Industries Predictions: 2

Avg Time Cost: 2.07 hrs

Show 10 entries

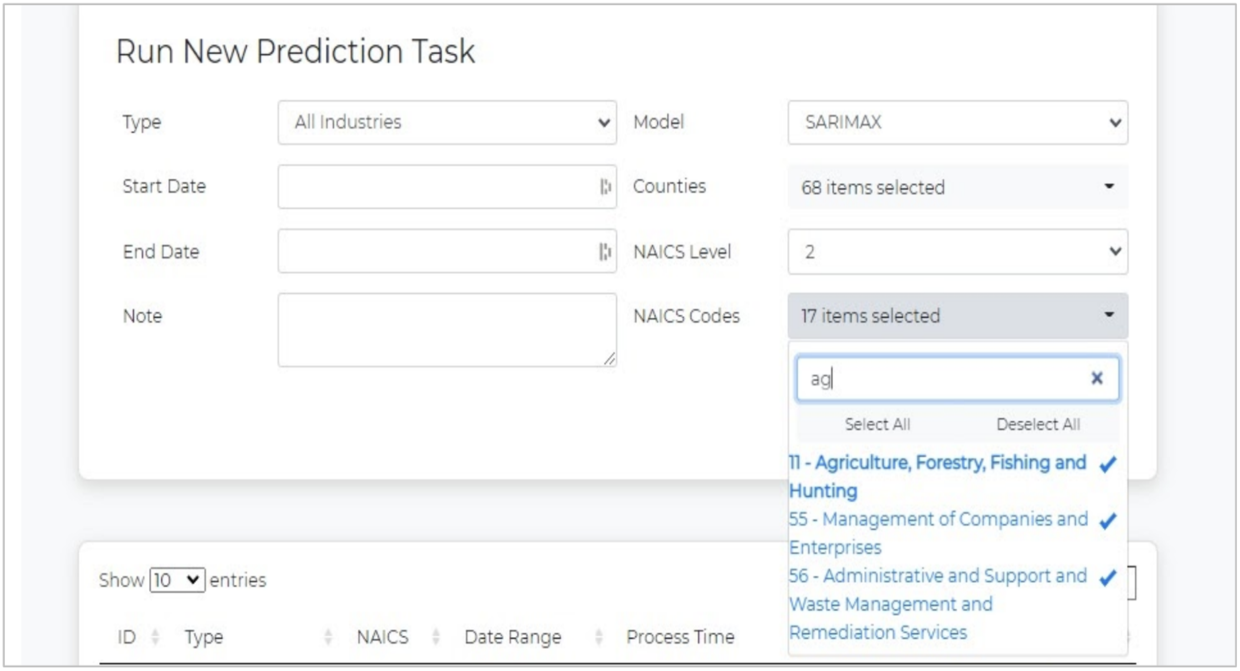
Search:

ID	Type	NAICS	Date Range	Process Time	Status	Log
4	All Industries	2	0001-01-01 9999-12-31	2021-10-21 14:27:23	Successful	Show Log X
1	All Industries	2	0001-01-01 9999-12-31	2021-10-18 20:53:43	Successful	Show Log X

Showing 1 to 2 of 2 entries

[Fig 5] New Predictions Page

New Predictions Page



Run New Prediction Task

Type: All Industries

Model: SARIMAX

Start Date:

Counties: 68 items selected

End Date:

NAICS Level: 2

Note:

NAICS Codes: 17 items selected

ag

11 - Agriculture, Forestry, Fishing and Hunting ✓

55 - Management of Companies and Enterprises ✓

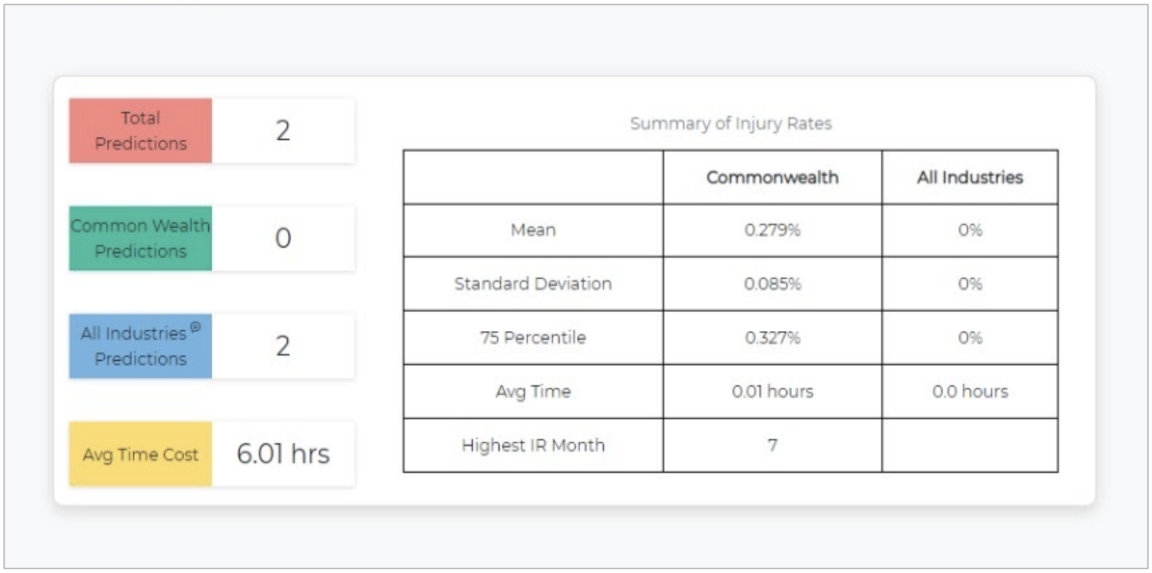
56 - Administrative and Support and Waste Management and Remediation Services ✓

Show 10 entries

ID	Type	NAICS	Date Range	Process Time

[Fig 6] New Statistics Page

New Statistics Page



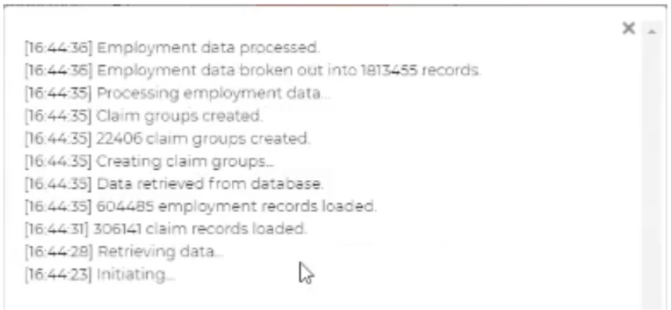
Total Predictions	2
Common Wealth Predictions	0
All Industries Predictions	2
Avg Time Cost	6.01 hrs

Summary of Injury Rates

	Commonwealth	All Industries
Mean	0.279%	0%
Standard Deviation	0.085%	0%
75 Percentile	0.327%	0%
Avg Time	0.01 hours	0.0 hours
Highest IR Month	7	

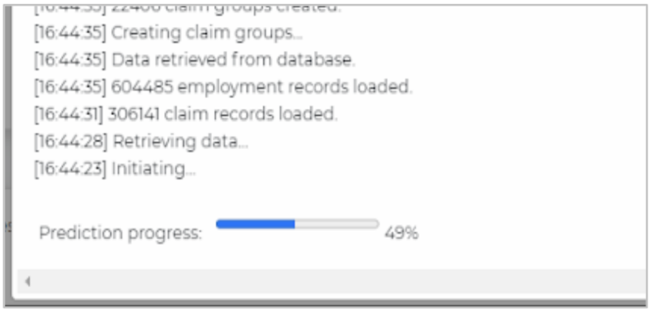
[Fig 7] Process Bar Changes

Old Prediction Model Log



```
[16:44:36] Employment data processed.
[16:44:36] Employment data broken out into 1813455 records.
[16:44:35] Processing employment data...
[16:44:35] Claim groups created.
[16:44:35] 22406 claim groups created.
[16:44:35] Creating claim groups...
[16:44:35] Data retrieved from database.
[16:44:35] 604485 employment records loaded.
[16:44:31] 306141 claim records loaded.
[16:44:28] Retrieving data...
[16:44:23] Initiating...
```

New Prediction Model Log



```
[16:44:35] 22406 claim groups created.
[16:44:35] Creating claim groups...
[16:44:35] Data retrieved from database.
[16:44:35] 604485 employment records loaded.
[16:44:31] 306141 claim records loaded.
[16:44:28] Retrieving data...
[16:44:23] Initiating...

Prediction progress:  49%
```

Security Feature + Code Cleansing

Upon examination of the existing codebase, the CMU team identified several areas of improvements within the fields of compliance and security. Some issues that stood out were:

1. Presence of staging AWS credentials and hard-coded passwords in code
2. Lack of logging functionality and excessive use of print statements
3. Scattered global setting definitions
4. Lack of quality control in production codebase
5. Outdated deployment instructions

The team took the following actions to address these issues.

1. Added a SAST analyzer to detect security vulnerabilities
2. Refactored existing code to include a local logging functionality

3. Aggregate globally defined settings into a settings.py file
4. Established a best practice guideline for development
5. Add build tools and linters to detect syntax and format issues in each code push
6. Updated deployment user manual and help page

The addition of SAST analyzer and build tools prevents future developers from introducing credentials and errors into the codebase. Refactoring and aggregating setting definitions improves code readability and eases the ramp-up process for future developers. By adding logging functionality and updating the deployment user manual, we ensure an easy hand-off process to the client and make troubleshooting tractable. The actions taken enforces a more rigorous standard in the development phase, while lifting the heavy load for deployment and monitoring in the production environment.

Results and Recommendations

The CMU team was able to successfully port the beta application into a production Windows environment and make the application available for use. In addition, we removed all non-compliant services and software. As a result, our client can now utilize the application in their business processes.

Throughout this process, our team also made further improvements to the application. We performed a full code review of the prediction module code to ensure that the forecasts would be computed correctly. As a result, we identified errors in the code that were causing inaccurate forecasts. These errors were identified and corrected. Furthermore, the code was streamlined to make it faster and easier to read and understand.

At the conclusion of our project, we believe our client now has a functioning and accurate application that they can use in their business. This application will allow them to collect insights and make informed decisions on where to allocate their resources and training to hopefully reduce the number of workplace injuries and illnesses throughout Pennsylvania.

While the application can now be used for business, we do not believe this is the final solution. There are areas where further enhancements could be made to gain further insights and more accurate forecasts. For example, using more sophisticated modeling approaches and enhanced reporting functionality. These areas are discussed in more detail in the coming sections.

Lessons Learned

In this project the CMU team acquired the following skills:

- Coding Skills
  - Web Application Development with Django framework
  - GUI Design and Development
  - Security Development (logging)
  - Code cleaning with SAST/Linters

- Production Deployment Skills
    - Windows Server 2019 Configuration
    - Microsoft SQL Server Configuration
    - Microsoft IIS Web Server Configuration
- Software Development Skills
    - Project Management
    - Agile Methods
    - Quality Assurance and Risk/Issue management
    - Teamwork
    - Communication
    - Presentation

Suggestions for future works

Throughout the course of the project, the team identified a few areas where future improvements could be made to the application. These are areas our team would have liked to explore further given enough time and resources.

Reporting Enhancements

The application currently utilizes a Power BI report to evaluate the prediction outcomes. Our team feels that this functionality creates greater complexity and makes the reporting harder to maintain. We believe that this functionality should be implemented within the web application as a separate tab. That way the user does not have to navigate back and forth between different applications. This would also make the application more streamlined from a maintenance perspective as the code would be housed within the Django framework and would be updated as other pieces of the application are updated.

In addition, we believe that the features included in the report could be enhanced to allow users to better evaluate different prediction model outcomes. For example, it would be nice to see what the model's accuracy was for a given prediction. Furthermore, having the ability to compare models side by side would allow the team to better assess which model seems more reasonable.

Prediction Model Enhancements

Our team enhanced the application to allow for two different models to be used for predicting future injury rates: Prophet and SARIMAX. These models are autoregressive and forecast future injury rates based on historical injury rates. However, these models could be enhanced by adding exogenous variables that may help forecast injury rates such as technological or innovation markers. Additionally, you could assess other types of models. The code has been designed in such a way that allows for the easy addition of other models. Furthermore, the prediction process uses the most recent employment numbers to estimate future workplace incidents. However, the modeling process could be made more sophisticated by using forecasted employment numbers.