Heinz College, Carnegie Mellon University

Policy Pandas Project Report

95829 - Software Design for Data Scientists, Spring 2022

May 2, 2022

Kelly McManus, kellymcm
Ryosuke Nakashima, rnakashi
Sudheeshna Sampath, sudheess
Pawanjeet Singh, pawanjes
Koushik Subramani Murali, kmsubram

Contents

Project Charter

Business background

Alleghany County has a $109M budget for mental health resources each year. Many of these resources are used for early education about mental health or crisis intervention and treatment (Allegheny County Department of Human Services, 2019). But, are we certain that this budget is having an impact on the local community? Research shows that there is an implementation gap for mental health policies where many individuals are not getting the help they need (Campion, 2018). This is one of the biggest challenges faced in improving public mental health. This is often due to a lack of local assessments conducted to identify local areas that suffer from the implementation gap.

Therefore, the business problem we address is the ability to identify counties in the United States that are lagging in quality of mental health and identify key factors that contribute to their mental health levels. Our solution aims to help county officials prioritize mental health policies to make the necessary allocation and funding of health care resources that will improve their county's gap in mental health.

Our anticipated clients are county health officials in the United States. They are in the public health domain. They will be able to use our solution to compare their county to national averages and understand which areas are negatively impacting their county's mental health. In addition, we expect our solution to contribute to public health knowledge for citizens concerned with the domain as a secondary outcome.

Scope

- *Solution* - The data science solution we build will provide the following features:
    - Identify top health factors that impact mental health
    - Compare the factors to the national averages
    - Provide recommendations for county improvement areas

  County officials will utilize these insights from our solution to drive positive changes in different counties and improve the overall mental health of the county.

- *Project Activities* - The project follows practices from CRISP-DM and Agile. We will develop a data science solution for the business problem using Python's data science libraries and deploy the solutions to Heroku (PaaS) as a web application using Tableau's user-friendly visual presentations.

- *Customer Channels* - The customers will be able to access the solution through the Internet browser.

Personnel

- *Clients* - The clients are the federal health departments and local health departments, including but not limited to the following.
    - Federal Health Departments
        - U.S. Department of Health & Human Services
        - Substance Abuse and Mental Health Services Administration

    - Local Health Departments
        - Pennsylvania Department of Health
        - Allegheny County Health Department

- *Project Team* - The project team forms a Scrum team. Team members and roles are as follows.

Scrum Role	Name
Team Member - Data Engineer	Sudheeshna
Team Member - Data Scientist	Koushik
Team Member - Data Scientist	Pawan
Team Member - Data Engineer	Kelly
Team Member - Software Engineer	Ryosuke

Metrics

To measure the success of the project, we have defined both quantifiable and qualitative objectives.

- *Quantifiable Objective* - The “mental health distress index” (0-1 range) is the primary indicator of the general mental health status of a county. A high value of this metric indicates that there is high mental distress in a particular county. This value will be our leading indicator to understand if policies developed from our product’s recommendations are, in fact, helping to reduce mental health distress in the county.
- *Qualitative Objectives* - The mental health distress index is driven by certain underlying health-related factors that we can uncover using exploratory data analysis and machine learning techniques. Identifying these top factors that contribute to driving the mental health distress index is essential in identifying the focus areas for implementing new policies to improve mental health at a county and state level.

The objective of the new policies implemented after identifying the top contributors to mental health is to reduce the mental health distress in a county. Observing a statistically significant decrease in the mental health distress index value prior to implementing new policies and observing the change in the mental health distress index over a time frame suggests that the implemented policies are effective.

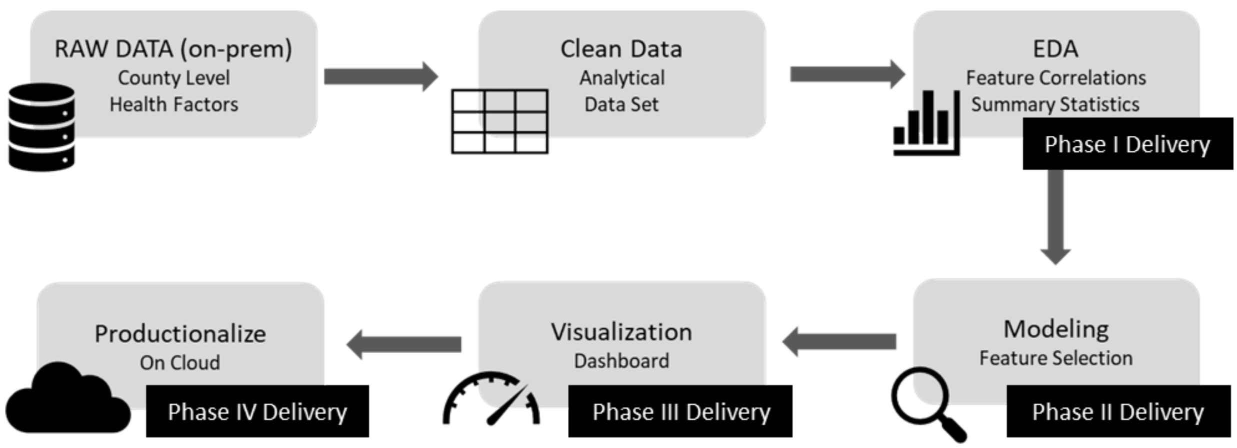
To measure the success of the policies, we plan to record the mental health distress values at a particular date as a baseline and re-evaluate the metric after policies have been implemented after a 6-month timeframe.

Plan

To properly plan and execute the project, we have broken up our project into four phases. These phases are visualized in Figure 1:

Phase		Deliverables
I	Exploratory Data Analysis and Feature Engineering	● Data preparation and analysis report consisting of summary statistics and feature correlations (Jupyter Notebook).
II	Modeling and Feature Selection	● Model performance report and top features selected for final tool development (Jupyter Notebook). ● Model deployed locally with Flask virtual environment.
III	Business Intelligence tool MVP	● Proof of Concept Tableau dashboard connected to model (Python Notebook, Dashboard Mock-up) ● Tableau dashboard embedded into HTML page locally with Flask virtual environment.
IV	Deploy to Staging	● Model and app pushed to GitHub. ● Model and app deployed on Heroku. (Data, Code repository, EDA reports, Dashboard)

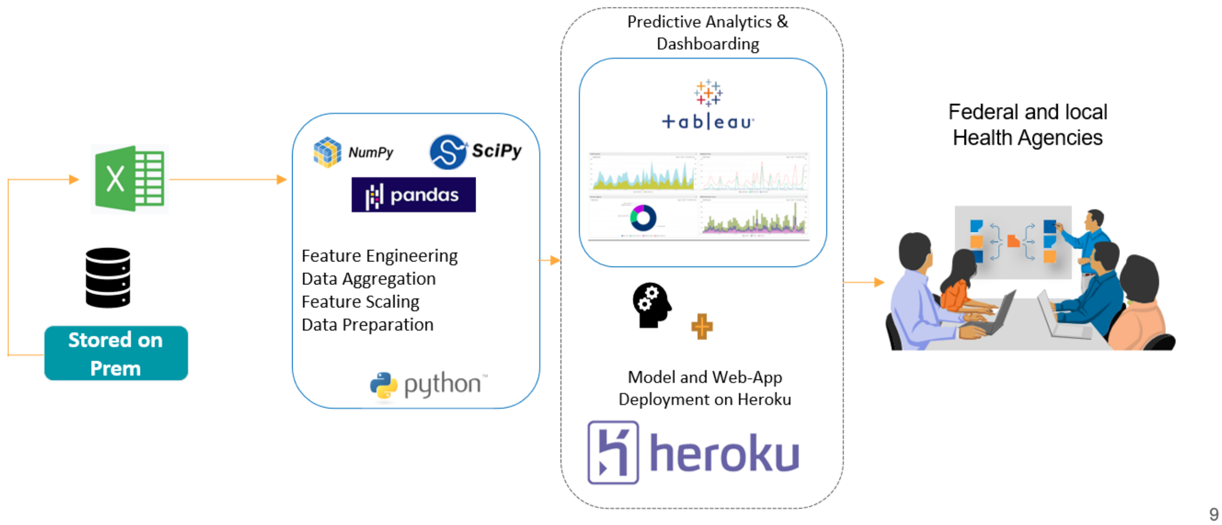
Figure 1: Overview of the Project Phases



Architecture

The product architecture flow is outlined in Figure 2 and is described in more detail below.

Figure 2: Overview of Architecture



- *Data Collection and Storage* - The raw health factor values are collected from health providers in the form of comma-separated value (.csv) files and stored on-premise.

- *Data Transformation and Wrangling* - Python libraries will be used for feature engineering, data aggregation, and exploratory data analysis to produce a final clean dataset. In particular, we will use the Pandas, NumPy, and SciPy libraries.

- *Modeling* - Again, python libraries will be used to build a prediction model and identify important features. In particular, the machine learning libraries Sklearn and Imblearn will be used.

- *Business Intelligence Tool* - The model will be connected to the visual analytics tool Tableau which will enable stakeholders to conduct self-serve analyses and arrive at data-driven policy decisions. Tableau will be embedded into our application.

- *Deploying* - Our model and Tableau-integrated app will be deployed to a Heroku server. The app will call the model and display the values in the Tableau dashboard.

- *Customer Using the Product* - Decision-makers will use the business intelligence tool to observe trends and narrow down the top contributors to mental distress in their respective communities to implement policy changes.

Communication

Our team will keep in touch using three different channels of communication. We are using a dedicated Slack channel to communicate on a daily basis. A shared Google Drive is being used, which acts as a repository to store all the data files, project charter, and modeling files. We are also using Trello to visualize progress in Kanban style, highlighting the weekly Sprint agenda, work in the pipeline, work in progress, completed tasks, and resources.

Our team is holding three “stand-up” style meetings every week. The current sync-ups are set for Wednesday, Friday, and Sunday evenings. Sprint Planning will be conducted every Sunday, to finalize the objective, (or goal of the week-long Sprint), work tasks, and scope for the subsequent week.

Data Report

Data Summary

The data we employ in our model is sourced from the County Health Rankings website. The County Health Rankings & Roadmaps is a program of the University of Wisconsin Population Health Institute that works to improve health outcomes for all and to close the health disparities between people with different factors that influence one's health. The dataset contains descriptive and measured values about health factors and health outcomes for all County Health rankings ranked and unranked measures for each county in the United States.

The dataset used for our analysis can be found at the following link: https://www.countyhealthrankings.org/sites/default/files/media/document/analytic_data2021.csv

We employ this dataset to find top health factors that impact mental health and how they compare to the national average to build recommendations for policymakers. The insights from our model can be used to drive positive changes in different counties and improve the overall mental health of the county.

The dataset is loaded into the directory where our python file is located. We use a Jupyter notebook to perform EDA and modeling. We then shift the data frame created with the recommendations/ insights generated by our model to Tableau. The report is then deployed through Heroku where policymakers can view recommendations for improving health factors in their respective counties.

Data Dictionary

Dimension columns

- statecode - FIPS state code
- countycode - FIPS county code
- fipscode - 5-digit FIPS Code
- state - State abbreviation
- county - County or state name
- year - Release year
- county_ranked - Indicator variable noting whether or not the county was ranked (0=unranked, 1=ranked); this variable is set to missing for state and national level data, since these are not eligible to be ranked.

Measure columns

For the purpose of our model, we will be predicting a measure of mental distress in each county. The target variable is the column for "Frequent mental distress". The value depicts the ratio of the population that faces frequent mental distress.

Features

The following measures are recorded as a ratio of the population that experienced or has experienced the following. The data has information about general health, safety, education, income, clinical access, and other factors that might influence our target variables.

- Ratio of population using tobacco:
  - Adult smoking
- Ratio of people with the following diet and exercise problems:
  - Adult obesity
  - Food environment index
  - Physical inactivity
  - Access to exercise opportunities
  - Food insecurity
  - Limited access to healthy foods
  - Insufficient sleep
- Ratio of people with alcohol & drug abuse problems:
  - Excessive drinking
  - Alcohol-impaired driving deaths
  - Drug overdose deaths
- Ratio of people with sexual activity issues:
  - Sexual Activity Sexually transmitted infections
  - Teen births
- Ratio of people with access to the following clinical care:
  - Primary care physicians
  - Dentists
  - Mental health providers
  - Mammography screening
  - Flu vaccinations
  - Uninsured adults
  - Uninsured children
- Measure of people with following socio-economic factors (in ratio):
  - High school completion
  - Some college
  - Unemployment
  - Income inequality
  - Social associations
  - High school graduation
  - Disconnected youth
  - Reading scores
  - Math scores
  - Residential segregation - Black/White
  - Median household income
  - Residential segregation
- Ratio of the following in area of residence:
  - Violent crime
  - Injury deaths
  - Homicides
  - Suicides
  - Firearm fatalities

- Juvenile arrests

- Measure of quality of the area of residence:
    - Air pollution - particulate matter
    - Drinking water violations
    - Severe housing problems
    - Driving alone to work
    - Long commute - driving alone
    - Traffic volume
    - Homeownership
    - Severe housing cost burden

Data sources

Raw Data Source

This dataset provides records for each county in the United States and their measured health factors. These measures are calculated from a variety of federal and local data sources. The initial raw dataset has 3194 counties with 690 features.

Processed Data

To ensure the model is learning from good quality data, we pre-processed the data using the following data cleaning and preparation techniques with Jupyter notebooks to narrow down the dataset to 2616 counties with 14 columns.

- *Dimensionality reduction* - The original dataset has 690 features. For our model, we will restrict our data to only use the "health factors" raw value features. These features are the recorded values for counties that will train the model to predict mental distress. After removing all unnecessary columns, the dataset has 34 features remaining.

- *Inputting missing values* - Some of the records have missing values in their dataset. To handle these values, we imputed the county's missing values with the median column value for the state that the county is in. For example, if Alleghany County has missing values in the "Physical inactivity raw value" column, it would be imputed with Pennsylvania's median value for "Physical inactivity raw value". After the imputation was conducted, if a column still had missing values, the column was removed. Our model will require all values to be filled for all counties and if the remaining values were not filled during imputation, it indicates that a state does not record these features and therefore the feature was removed.

- *Removing outliers* - In the current state, our dataset had outliers. To ensure a generalized model, we removed outliers by taking the Z-Score of each value in the dataset. If the Z-score was above 3 standard deviations, the record was removed from the analysis.

- *Exploratory data analysis* - To better understand the relationship between different features and assess the cleaning technique we used histograms and heatmaps to visualize the data.
    - Histograms - Plotting the histogram for all numeric variables, we see that our data imputation and handling of outliers provided promising results as the histograms appear

to be heading towards a normal distribution. Some plots still are partially skewed, but for our first iteration of the model, it is okay for the data to have a few outliers.
    - Correlation - Plotting a heatmap, we see the strength of the correlations between our outcome variable and features that are used to simplify our model in feature selection.

- *Feature selection* - The dataset still holds 33 features. To create a simple model that we can use as an MVP, we will narrow this dataset down to 10 features. We use correlation values to select the top 10 features that are strongest correlated with our label, 'Frequent mental distress raw value'. The remaining top 10 features are discussed below in the Final Dataset section.

- *Data transformation* - Lastly, we split the data into y and X data frames and use a Standard Scalar to normalize the X data so that our model will not be skewed by particularly large coefficients.

Final dataset

After applying the data cleaning and preparation techniques, we are left with 2616 county records and 14 columns: 3 descriptive columns, 1 target, and 10 features.

- *Target variable* - Our target and prediction variable is "Frequent mental distress raw value".
- *Descriptive variables* - Three descriptive variables are kept in the final data frame to use as identification for the counties: "5-digit FIPS Code", "State Abbreviation", and "Name.
- *Feature variables* - After feature selection, we are left with the top 10 correlated features with our target variable. They are displayed in decreasing rank order below. These will be the 10 features used in our first model to predict mental distress in a county.

Feature rank	Feature name
1	Adult smoking raw value
2	Some college raw value
3	Children in poverty raw value
4	Excessive drinking raw value
5	High school completion raw value
6	Teen births raw value
7	Physical inactivity raw value
8	Food environment index raw value
9	Unemployment raw value
10	Injury deaths raw value

Model Report

Analytic Approach

The goal of our model is to predict the frequency of mental distress for a particular county given certain feature inputs. This is our target. The predicted value will be between 0 and 1. A value closer to 0 indicates a county has a low percentage of the population with mental distress, while a value closer to 1 indicates a county has a high percentage of the population with mental distress.

To predict this value, our model takes the following inputs:

- Adult smoking
- Some college
- Children in poverty
- Excessive drinking
- High school completion
- Teen births
- Physical inactivity
- Food environment index
- Unemployment
- Injury deaths

With the above values inputted by the consumer, we will call the predict function of the model to predict a value for the frequency of mental distress.

We explored three types of models to select the best one for our use case: Linear Regression, kNN, and XG Boost.

Model Exploration

For the three models we explored, we divided the data into training and test datasets. Training data for the model to learn and test data for the model to predict with unseen data in order to assess accuracy.

Model exploration can be found in this Jupyter Notebook.

Linear Regression

A linear regression model fits an ordinary least squares linear model to the dataset with the goal of minimizing the residual sum of squares. Our linear regression model uses default values as we will be using it as a simple, baseline prediction.

We checked for features with high VIF values to detect collinearity between the outcome variable and the features. No features have collinearity detected, therefore no features were removed.

The evaluation metrics for the linear regression model resulted as follows:

R-squared	0.8896
Mean absolute percentage error	0.0422
Root mean squared error	0.0077

The linear regression performed surprisingly well on the dataset. If we are to use a different model, it would have to outperform the linear regression baseline model.

kNN Regression

The k-nearest neighbor regression, is a regression model that averages the k-nearest training points around a test value to determine the predicted value. This is a non-parametric model as all training and learning of the model is done at test time. This can result in this model being slow to predict a value which is not ideal for production environments when you want to get a predicted value fast.

The kNN regression has one core hyperparameter that needs to be tuned: n_neighbors. This hyperparameter determines the number of neighbors that should be averaged to give the most accurate prediction. Using 5-fold cross-validation and grid search with neg_root_mean_squared_error scoring parameter, the tuned value is n_neighbors = 12. We moved forward to predict with this hyperparameter value.

The evaluations metrics for the kNN regression model resulted as follows:

R-squared	0.9201
Mean absolute percentage error	0.0293
Root mean squared error	0.0056

The kNN regression model showed to be improving in predictive power over the baseline linear regression, but because of the drawbacks of using this model in a production environment, we decided to explore other models.

XGBoost Regressor

XGBoost in an ensemble regression model that simultaneously reduces variance and bias. The model uses weak learners to sequentially train and weight models to build a final aggregated model. In boosted regression, the learned models fit to residuals to determine model and data weights.

Similar to kNN regression, the XGBoost regressor has hyperparameters that need to be tuned. Using 5-fold cross-validation and grid search with neg_root_mean_squared_error scoring parameter, the best estimator results in the hyperparameters learning_rate = 0.15, max_depth = 3, and n_estimators = 90.

The evaluations metrics for the XGBoost regressor model resulted as follows:

R-squared	0.9513
Mean absolute percentage error	0.0231
Root mean squared error	0.0043

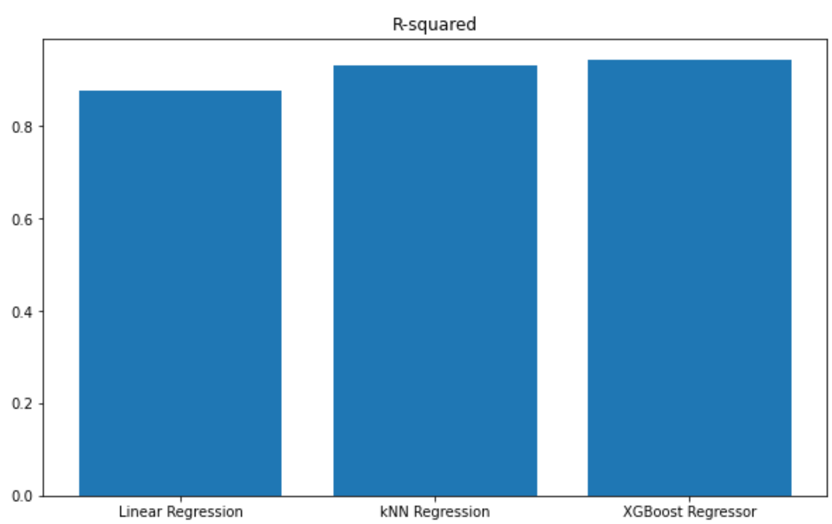
The XGBoost model seems to perform better than the prior two models. This can be seen the best when we plot the metrics to compare model performance.

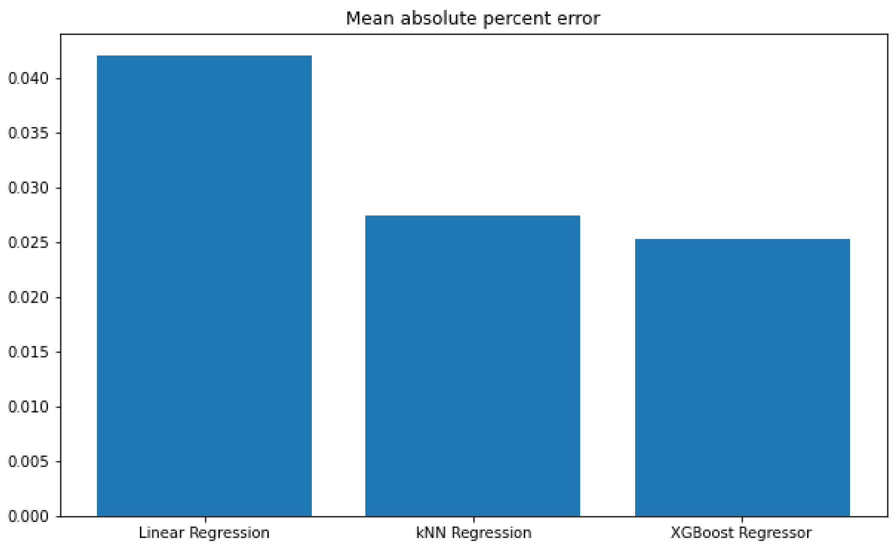
Model Performance

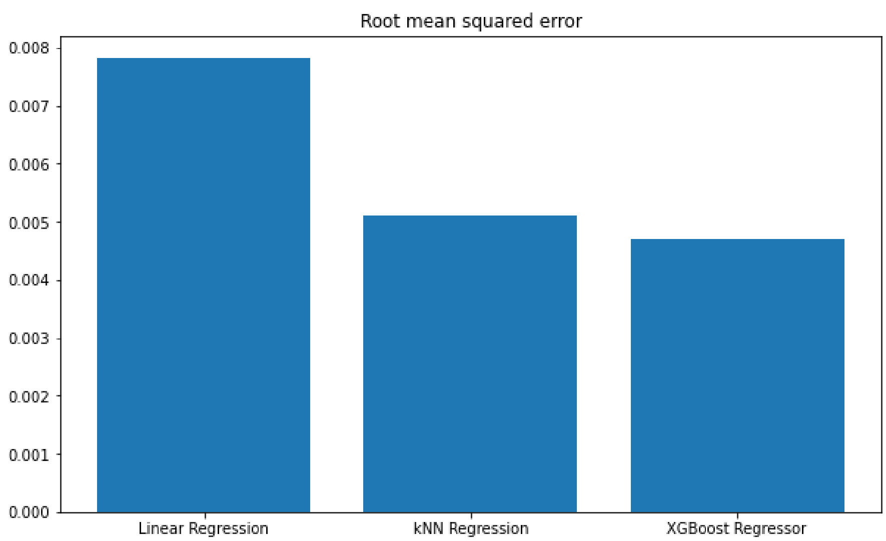
We used three evaluation metrics to compare the performance of our proposed models:

- R-squared - Percent of target variable that is predicted by the features. The goal is to have this value closer to 1.
- Mean absolute percentage error - Absolute percentage loss comparing actual and predicted values. The goal is to have this value closer to 0.
- Root mean squared error - Square root of the mean squared error accuracy estimator. The goal is to have this value closer to 0.

The charts below compare each model with these three performance metrics:





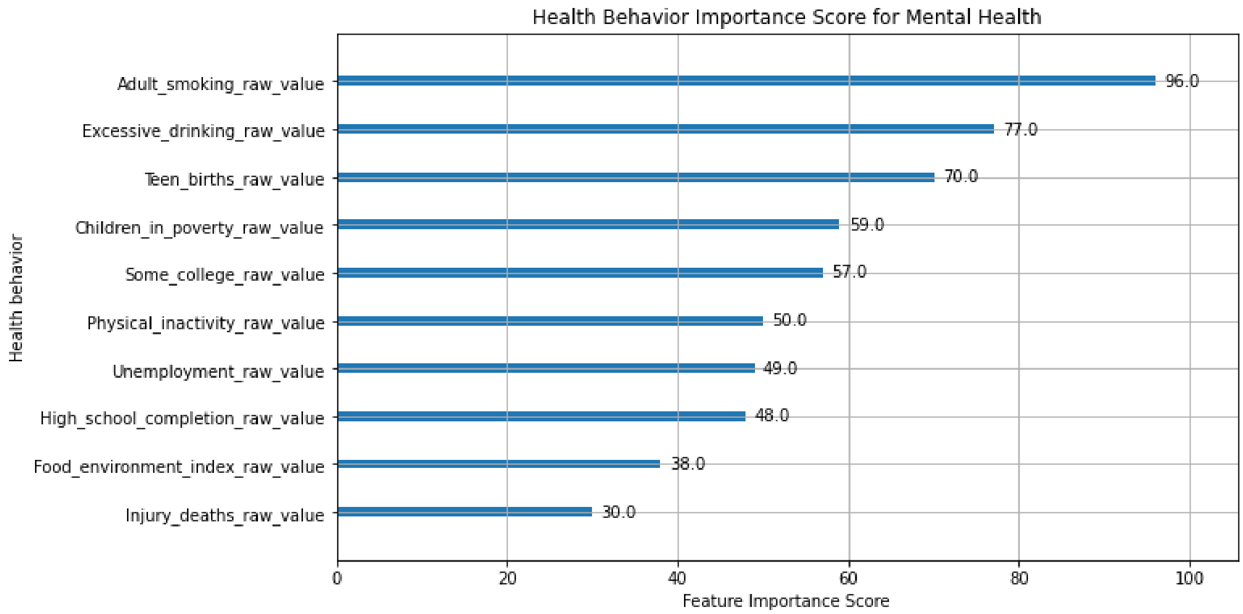


Across all three performance metrics, the XGBoost model shows to perform the best.

Model Understanding

After evaluating model performance, we decided to move forward with the XGBoost model as it will perform well in production and reduces model variance and bias to produce a nice prediction of frequent mental distress of a county.

Digging further into this model, we can extract the feature importance scores:



The feature importance chart ranks the features by weight. Weight is calculated by the number of times a feature is used in the boosted trees. We can see that Adult smoking is by far the most important feature in predicted frequent mental distress in a county. On the other hand, the food environment index shows to be the least important feature in the prediction. The feature importance chart values will be used to display to our customers which factor to keep in mind when hoping to improve mental distress.

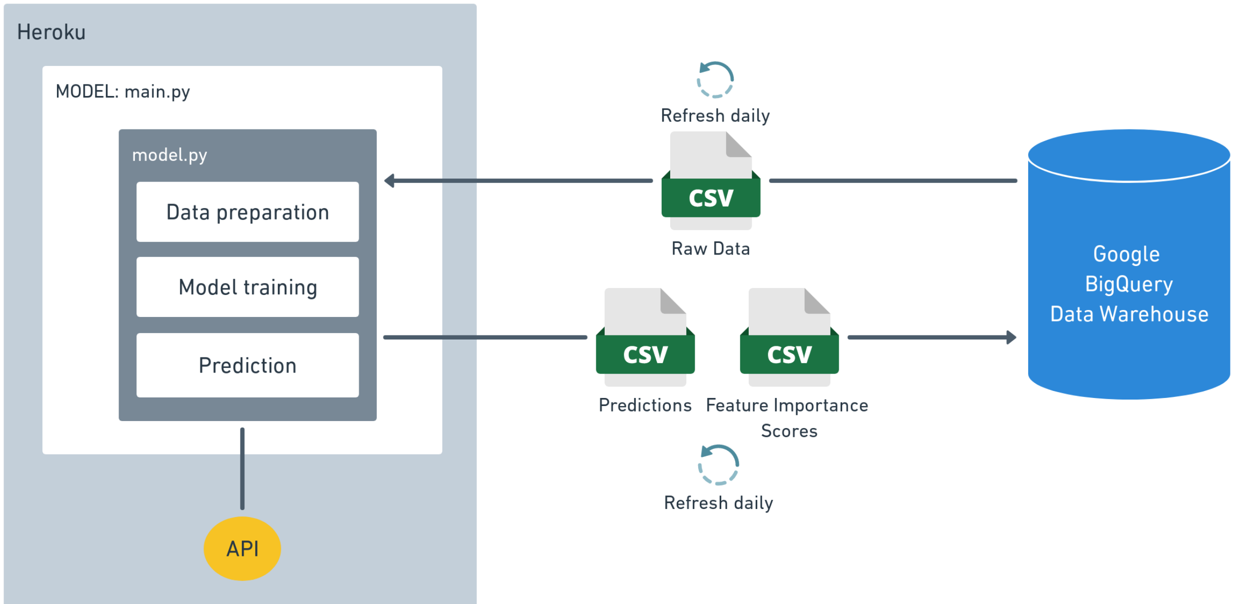
The feature importance values are exported to a CSV to be consumed by Tableau and displayed to the user.

Model Conclusions

After training and evaluating linear regression, kNN regression, and XGBoost regressor, we will be moving forward with the XGBoost model as our minimal viable model to predict frequent mental distress. In later iterations of the project, the model will be iteratively improved on to ensure high accuracy and performance.

Overall, our model pipeline is displayed in Figure 3. The model.py file consumes our raw analytics_data2021.csv dataset. After data cleaning and preparation, it will train the model and predict the target. Two CSV files are generated, feature_importance and predictions, which are both updated at a scheduled interval and stored in Google BigQuery. The model.py file will be hosted on Heroku and have an API that will be able to be called by the web application to predict the target with a given set of parameters.

Figure 3: Model Architecture



Solution Report

Our final solution architecture uses the XGBoost model and applies a front-end user interface to analyze county mental health factors and make mental health distress predictions.

Solution Architecture

To set up our solution, we use the following tools:

- Github - Store code base and connects to Heroku server.
- Heroku - Hosts the model and application and is integrated with Github to run code in repositories.
- Google BigQuery - Data warehouse that stores raw data files and
- Tableau - Visualizes data from BigQuery in an interactive dashboard.

With the combination of these four tools, we were able to implement a software architecture that enables county officials to analyze their county's mental health status and understand key factors impacting mental health.

The final solution can be accessed here (Public): https://policy-pandas-web-app.herokuapp.com/

Github repositories for the codebase can be found here (Private):

- Model Repository: https://github.com/RyosukeNakashima/policy-pandas-model
- Application Repository: https://github.com/RyosukeNakashima/policy-pandas-app

Figure 4 describes the software architecture and flow of data throughout the pipeline.

Google BigQuery

Google BigQuery is used to store and refresh our datasets. The original, raw dataset, analytics_data2021.csv, is stored in BigQuery. Every day at midnight, a scheduled run of the model.py file is triggered to refit the model with new data. The feature_importance.csv and predictions.csv files are output and stored back in BigQuery to be used by Tableau.

Tableau

We decided to use Tableau as the front-end for self-serve analytics for our customers. Tableau reads the original data values, feature importance, and county predictions from BigQuery and updates the pre-built dashboard with these values. We embedded the Tableau dashboard into the HTML of the application to provide our customers with easy access to analyze the mental health factors.

Github

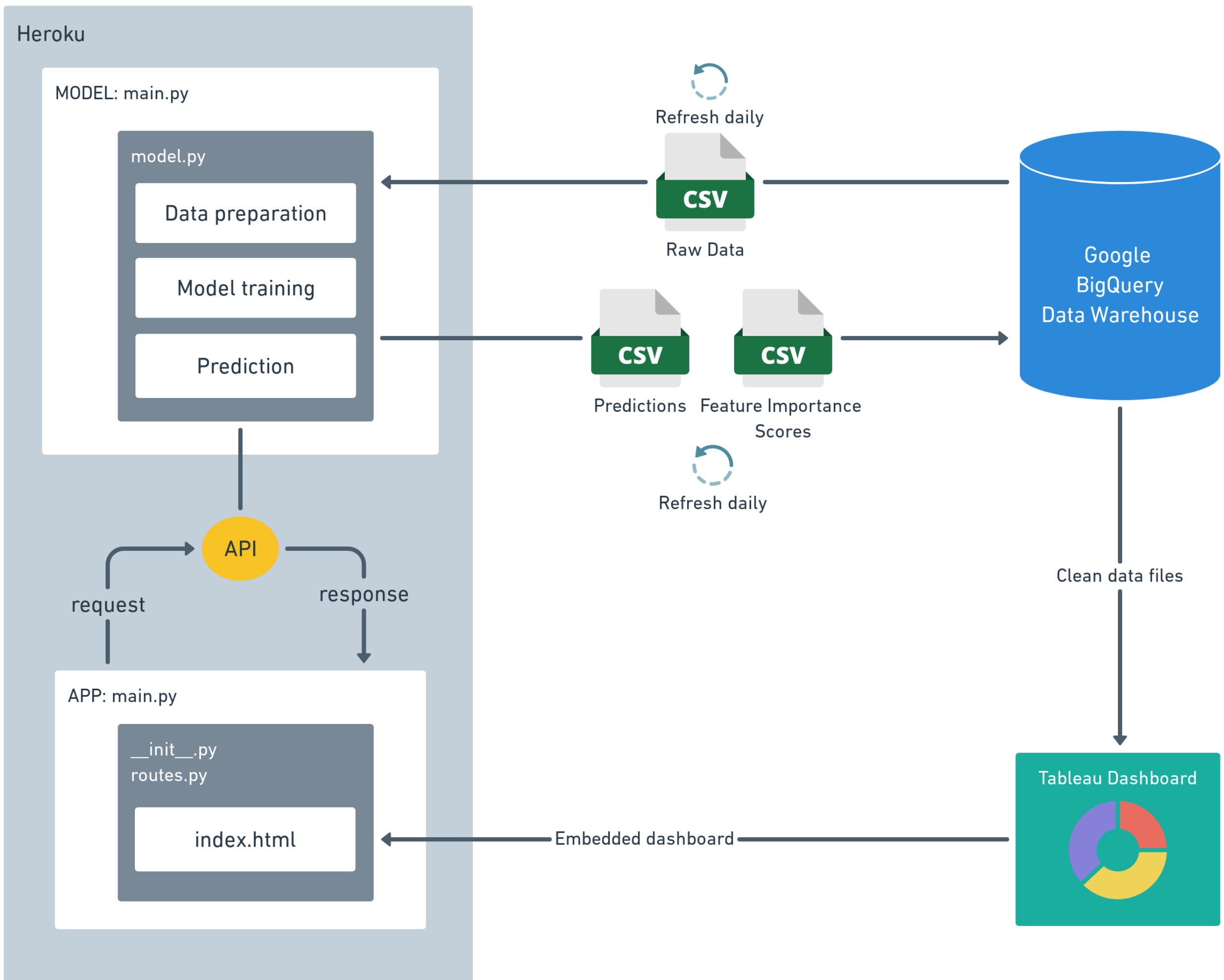
The prediction model was converted to a model.py file that is pushed to a Github repository to host the codebase. This contains a main.py file that points to the model.py file to run the API and predict() function.

Similarly, the application has its own repository on Github to store the application frontend, HTML, and backend, Python, files.

Heroku

In Heroku, there are two pipelines for the model and application. These pipelines pull the codebase from Github and deploy the application on the Heroku server. The model and the application interact with each other when the user submits a request to predict mental distress. The application sends a request with the user input in JSON to the model API. The model then runs the predict() function and returns the response prediction to the application to display.

Figure 4: Solution Architecture



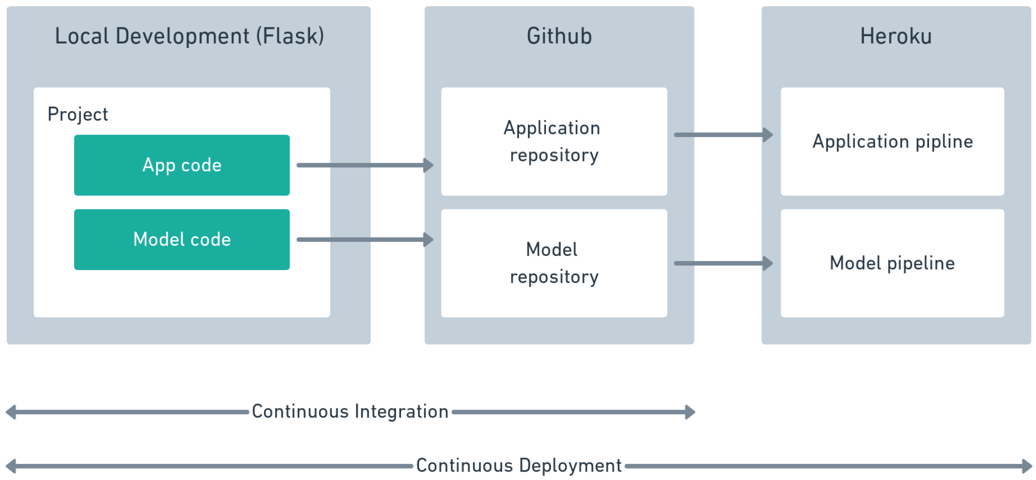
Development Architecture

Our solution provides a small continuous deployment pipeline from our local development environment to the Heroku production server. Figure 5 shows the development architecture flow.

Locally, our codebase is stored in a Project folder where we run a Flask virtual environment to develop and test the application and model. The Project folder is connected to a GitHub repository to enable version control and testing. The GitHub repository is then integrated with Heroku to host the application on the Heroku server. This flow enables a simple update flow of the application and model from local development to production.

We do not currently have a staging environment. To enable full continuous integration, delivery, and deployment we should add a staging environment and testing at each stage.

Figure 5: Development Architecture



Exit Report

Overall, the goal of this project is to identify counties in the United States that are lagging in quality of mental health and provide key factors that contribute to their mental health levels. After conducting data preparation and developing a model that identifies top mental health factors and predicts mental health distress, we were able to deliver a minimal viable product of the county mental health analysis tool. These phases of development are explained in detail in the Data Report, Model Report, and Solution Report.

Customer Benefits

Uncovering information on mental health levels and their contributing factors will help county officials prioritize mental health policies and allocate funding towards resources that will address the core factors impacting mental health. Our core metric to track the success is the change in the mental health distress level of a county over time. This metric will be impacted by new policies that are put in place by the policymakers who use the dashboard and prediction tool.

While at this time it can not be quantified exactly the impact that our tool will have on mental distress, we can be certain that it will help save time in budget planning and allocation. We hope to see the impact of this tool within counties over the next year and see a significant decrease in mental health distress nationwide based due to more target policies and better allocated mental health resources.

Project Learnings

This project was the first time our project team worked to operationalize a data science model. With the first-time doing anything, there are many learnings which are listed below.

Data Science / Engineering

- *Build a lightweight model* - Our model fits on training data when the web service is first loaded. Our first iteration of the model included some heavy cleaning and transformation tasks that caused issues when deploying the model to Heroku, which has some restrictions (detailed in the next bullet point). We learned that we need to connect as much cleaning and training prior to hosting the model to ensure that the production runs smooth and quickly for the user to get a prediction response.
- *Deploying to Heroku* - We faced challenges running the model on Heroku though it successfully ran locally within a virtual environment. As a result of research, we found that Heroku has the following rules: (1) a web process of Heroku, called “web dyno,” has to be launched within 60 seconds, and (2) web dyno has to respond to an HTTP request within 30 seconds. If the conditions are not met, Heroku forcibly kills the process and retries it, leading to repeated failure of the same process. In our design, data preparation and model training are coupled with starting a web dyno, and model prediction is coupled with the web dyno’s response to an HTTP request. Since our original model was relatively heavy, we breached the timeframe above, and the processes never got completed. To overcome these challenges, we made the model lighter to complete the web dyno’s processes within the timeframe. In addition, we found that Heroku runs two web dynos as default to secure redundancy, making the processes further heavier by performing data preparation, model training, and model prediction twice. Therefore, we changed the configuration so that Heroku runs only one web dyno. Our current design is acceptable as MVP but may not be scalable enough to be a production system. To address the

bottleneck, we will need to decouple data preparation, model training, and model prediction from web dyno's starting and response processes in the future. We can realize such design by making them back-end processes using a queuing mechanism.

- *Staging* - Our current pipeline does not benefit from a staging environment to test integration with the environment and get user feedback. We believe adding a staging environment would ensure smoother deployments to production and enable continuous delivery.

Product

- *Understand the customer* - At the start of the project, we had a hard time understanding the goal. Discussing directly with the customer and conducting customer interviews would have helped to clarify the goal and gather user requirements.
- *Visualize the flow* - When discussing different product flows or software architectures, we found that it was useful to visualize the flow to ensure that the team was on the same page. We made use of Whimsical to build flow diagrams and wireframes to experiment with possible scenarios and draft dashboard designs before implementing them into the product.

Next Steps

We hope this is not the end of this project. In the next steps, we plan to complete the following tasks.

- *Testing & Feedback*: Put our product in the hands of county officials to gather customer feedback and implement recommendations into further iterations of the product.
- *Measure:* Quantify the impact of our product overtime to ensure the continued investment in the project will deliver value to the counties and achieve project goals.
- *Expand*: Research possible expansion of the project into other domains. Potentially in education or transportation, where we can measure a county's quality index and recommend factors to focus better on their budget and resources.

Overall, this is just the start of the project. We hope to see a quantifiable impact in a reduction of mental health distress at a county level within the next few years.

References

Allegheny County Department of Human Services. (2019). *DHS Office of Behavioral Health.* https://www.alleghenycountyanalytics.us/wp-content/uploads/2020/08/Office-Profiles-OBH-2020.pdf

Campion J. (2018). *Public mental health: key challenges and opportunities.* BJPsych Int. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6690256/