

Group 5

Module 6

Group Project: Final Project

Class ALY6150.81185.202335 – Healthcare/Pharmaceutical Data and Applications

Ashutosh Singh, Aytaj Khankishiyeva, Mehdi Mammadov

# TABLE OF CONTENT

[Introduction 3](#_Toc138947885)

[Descriptive Analysis 3](#_Toc138947886)

[Data Cleaning and Preprocessing 4](#_Toc138947887)

[Exploratory Data Analysis 5](#_Toc138947888)

[Data Manipulation 13](#_Toc138947889)

[Feature Engineering 13](#_Toc138947890)

[Predictive Analysis 14](#_Toc138947891)

[Conclusion 19](#_Toc138947892)

[Reference 20](#_Toc138947893)

**Introduction**

This assignment is about disease in the United States, which has 50 states in 2019-2020, with different categories and metrics. The project is built on two separate datasets: the first contains various estimates, chronic illness for various age groups, states, population, and health categories, whilst the second contains variables such as states, race, ten-year age groups, and population within percentage.

The first stage in assignment is to describe the variables, their kinds, and the number of missing values in the datasets. Then, using Exploratory Data Analysis (EDA), explore the data and extract all possible insights to uncover trends or patterns on the dataset, which also aids in the construction and development of models for decision making.

Our project will analyze disease prevalence in the United States between 2019 and 2020 across different groups and measures, identify patterns and trends, and build a predictive model using Random Forest and Deep Neural Networking models to forecast disease prevalence based on various factors. As a consequence of the investigation, we will establish which factors cause sickness in the United States. These insights and predictions can potentially contribute to decision-making processes related to public health interventions, resource allocation, and policy development.

Python was utilized for Descriptive Analysis and EDA. Let's go through the entire procedure step by step:

**Descriptive Analysis**

As the first step, we uploaded the necessary the packages like pandas, matplotlib. pyplot and seaborn in Python and then read first dataset which is a CSV file named ‘PLACES\_\_Local\_Data\_for\_Better\_Health\_\_County\_Data\_2022\_release.csv' into a Data Frame that is called as ‘data’. The second dataset which contains Notes, States, State Code, Ten-Year Age Groups, their code, race, race code and population attributes. Both databases contain various sorts of information, with the one focused on health measurements and the second on population demographics. They have the potential to be utilized for monitoring health trends, examining population characteristics, and performing health and demographic research.

Then we began to descriptive analysis from observing the first five rows of the first dataset using head() function get five rows of 21 columns (Year, StateAbbr, StateDesc, LocationName, DataSource, Category, Measure, Data\_Value\_Unit, Data\_Value\_Type, Data\_Value, Data\_Value\_Footnote\_Symbol, Data\_Value\_Footnote, Low\_Confidence\_Limit, High\_Confidence\_Limit, TotalPopulation, LocationID, CategoryID, MeasureId, DataValueTypeID, Short\_Question and Geolocation). Then we define the feature and categories of the first dataset which helped us to observe which columns are useful for our analysis. We printed the features with identical values or NaN (missing values) in the first dataset. We defined the ’DataSource’ and ‘Data\_Value\_Unit’ are identical values where ‘Data\_Value\_Footnote\_Symbol’ and ‘Data\_Value\_Footnote’ contains fully missing values. Additionally, using data.info () we obtained the summary of the first dataset which shows count of non-null values and their data types. We defined that Year, Data\_Value, Data\_Value\_Footnote\_Symbol, Data\_Value\_Footnote, Low\_Confidence\_Limit, High\_Confidence\_Limit, TotalPopulation, and LocationID are numerical values. Then, we described descriptive statistics of numerical columns in the first dataset using the describe () function. Before doing data cleaning process, the first dataset contained 188456 records with 21 columns.

After cleaning of second dataset, it contains 4 columns (States, Ten-Year Age Groups, Race and Population) with 2142 rows. When defining the feature and categories of variables, we noted that Race variable contains American Indian or Alaska Native, Asian, and other values.

We calculated the sum of Population variable for each category (State) in second dataset. For example, total population for Michigan is 8 005 854 while for North Dakota sum of population is 592 435 in this dataset.

**Data Cleaning and Preprocessing**

When we were doing descriptive analysis, we identified some the unnecessary columns which need to be removed from the data. Thus, we removed "DataSource","Data\_Value\_Unit", "Data\_Value\_Footnote\_Symbol","Data\_Value\_Footnote", "StateAbbr", "LocationID", "DataValueTypeID", "CategoryID", "MeasureId", "Low\_Confidence\_Limit’ and “High\_Confidence\_Limit” columns from the first dataset. Furthermore, ‘StatesDesc’ column contains all states name, but there was ‘United States’ values, we deleted these rows from ‘StatesDesc’ column. After little cleaning of the dataset, the first dataset consists of 188396 rows with 10 columns. In summary statistics of this dataset, we noticed that mean of Data\_Value is about 32 while maximum value of this variable is 93. The average of Total Population is about 104 556 while maximum value of Total Population is 9 943 046.

When we uploaded the second dataset, we removed the last 29 rows (notes) and the first column (Notes). So second dataset contains 2142 records with 7 variables. Additionally, there were States Code, Ten-year Age Groups Code and Race Code variables which were not useful for analysis; so, we decided to remove those unnecessary columns from the second dataset.

Next, we built bar chart for population by race which shows how total population is distributed over the races.

A picture containing line, plot, diagram, rectangle

Description automatically generated

Figure 1: Population by Race

From Figure 1, we observed that most of the people belong to White race; the Asian population is the second largest racial group in this dataset. Black or African American people also represent significant portion of this population.

We also established bar chart for Population by Ten-Year Age Groups which shows the how the people are distributed over the ages.

A picture containing plot, line, font, diagram

Description automatically generated

Figure 2: Population by Ten-Year Age Groups

From Figure 2, we noticed that all age groups have among total population. However the highest frequency is between 25 and 34 year old groups. From 65 years age, this frequency goes down in graph.

Moreover, we measured the population percentage for each state and added this calculated column to second dataset and removed Population column from second dataset. Now second dataset, contains 2132 rows with 4 columns. In that case, we can say that population percentage for Alabama states in 15-24 years old people who is Black or African American equals to 4.89%. After this, we transformed the original table to a new format where the unique combinations of 'States' and 'Ten-Year Age Groups' are the rows, the unique values of 'Race' are the columns, and the corresponding 'Population\_Percentage' values are populated in the cells.

**Exploratory Data Analysis**

Before performing exploratory data analysis, we checked counts of null values in each column of the ‘data’ Data Frame. In the result, we identified that there are no null values in any of the columns. For defining uniqueness of each value in ‘data’ table, we used unique () function and observed each unique values of each column from first and second dataset. For instance, ‘Data\_Value\_Type’ variable’s unique values are ‘Crude prevalence’ and ‘Age-adjusted prevalence’; ‘Category’ contains 'Health Outcomes', 'Prevention’, ‘Health Risk Behaviors’, and 'Health Status'; ‘Ten-Year Age Groups’ contains '15-24 years', '25-34 years', '35-44 years', '45-54 years', '55-64 years', '65-74 years’ and '75-84 years'. Additionally, we built histogram of Age groups to observe frequency of Ten-Year Age groups. As the result of histogram in Figure 3, we determined that all age groups are equally distributed in the second dataset.

A picture containing text, screenshot, line, parallel

Description automatically generated

Figure 3: Distribution of Age groups

Then, we built histogram for States to see which state have more frequency in compared to others.

A picture containing line, plot, font, screenshot

Description automatically generated

Figure 4: Frequency of States

In Figure 4, we defined that Texas has the highest frequency, then Georgia and Virginia are following Texas. The lowest frequency over the states are observed in Hawaii, Delaware, Rhode Island and District of Columbia.

Next, we created the histogram of Category from the first dataset and noticed that highest frequency occurred in Health Outcomes in Figure 5; Prevention is second category by passing 60 000. However, Health Risk Behaviors and Health Status categories have low records in comparison of Health Outcomes and Prevention.

A picture containing text, screenshot, line, plot

Description automatically generated

Figure 5: Frequency of Category

For defining relationship among variables in the first dataset, we built correlation matrix and pairwise scatter plots. Therefore, we noticed that there is medium negative relationship between Year and Data\_Value, but vert low positive relationship between Year and TotalPopulation in Figure 6. Furthermore, there is very low negative relationship between Data\_Value and TotalPopulation:

A picture containing text, diagram, screenshot, line

Description automatically generated

Figure 6: Relationship between variables in the disease data frame

Addition to this, we built pairwise scatter plots of second dataset:

A picture containing text, diagram

Description automatically generated

Figure 7: Relationship between variables in demographics data frame

From Figure 7, we could see that distribution of each race along the diagonal boxes. For example, it is clearly seen that White race group looks like normally distributed while rest of race groups are right skewed distributed. It is hard to say that between two races have relationship. We can see just low relationship between More than one race and Native Hawaiian or Other Pacific Islander races.

Furthermore, we grouped States for Total Population in the first dataset and created horizontal bar chart.

A picture containing text, screenshot, line, diagram

Description automatically generated

Figure 8: Distribution of population over States

From Figure 8, we can say that California, Texas, Florida, and New York have higher records among total population. There are less observation for Wyoming, Vermont, North Dakota, Alaska and so on.

Using x and y as longitude and latitude coordinates which are splits of ‘Geolocation’ variable; we converted these coordinates into map scales and built map using marker as ‘D’ and Color as ‘red’.

A picture containing map, drawing, illustration

Description automatically generated

Figure 9: Points of longitude and latitude

From Figure 9, we defined that those locations are belongs to USA states which has been highlighted by red color on the map.

Addition to the general map, for observing the density of disease expansion, we built ‘The Density of disease records over the US’ map.

A map of the united states

Description automatically generated with medium confidence

Figure 10: Intensity of disease over USA

From Figure 10, we could observe distribution of disease over USA within different colors which red shows high density while purple shows low density of disease. The map indicates that when east and middle states of the United States have high disease frequency, west states and around Alaska have low frequency.

A picture containing screenshot, map, colorfulness

Description automatically generated

Figure 11: The highest density in east coast of USA

In Figure 11, we highlighted some areas like Roanoke Rapids, Petersburg, Hopewell, Williamsburg- all these are in Virginia state, Manassas, Silver Spring, Towson which are Washington state areas as the highest density of illness records over the United States.

A picture containing screenshot, colorfulness, map

Description automatically generated

Figure 12: West coast of USA within lower disease records

From Figure 12, we identified that west coast areas have low density which purple color showed low disease records. Around these states high disease records have determined in some cities like Sacramento, San Jose, Salt Lake City, Colorado Springs, Utah, Bose (Idaho) which have colored with combination of green and yellow. Instead of having high population in California, we didn’t get high sickness over there.

We also created the chart by grouping Measure over the Total population in the following figure. From Figure 13, we can claim that there are similarity among measure among total population. Only ‘Taking medicine for high blood pressure control among adults aged>=18 years with high blood pressure’, ‘High cholesterol among adults aged >=18 years who have been screened in the past 5 years’, ‘High blood pressure among adults aged >=18 years’, and ‘Cholesterol screening among adults aged >= 18 years’ groups have less records in compared to the other measures.

A picture containing text, screenshot, document, receipt

Description automatically generated

Figure 13: Measure over the Total Population

Addition to this, we grouped total population into Category and created plot to observe how each category are grouped among total population in merged dataset.

A blue and white bar graph

Description automatically generated with low confidence

Figure 14: Total Population over Category

From Figure 14, we observed that Health Outcomes and Prevention are mostly occurred among Total Population according to the merged dataset.

**Data Manipulation**

For gaining meaningful insights and preparing data for further analysis, we manipulated data by renaming ‘StatesDesc’ column to ‘States’ column in second dataset before merging these two datasets. Before merging, we also dropped Population percentage from second dataset which we already have TotalPopulation variable from the first dataset. We merged the two data frames on the ‘State’ column. The merged table contains 7 898 832 rows with 12 columns: Year, States, LocationName, Category, Measure, Data Value Type, Data Value, Total Population, Short Question Text, Geolocation, Ten-Year Age Groups, and Race. In the merged table, Total Population was divided 100 for transforming from percentage to number.

However, the initial dataset provided information on cases rather than the population of specific cities. Therefore, in the second dataset, we computed the total population for each state and determined the proportion of each race within each state. Subsequently, we merged these two datasets using the state category as the basis. Essentially, we transformed each row for every state and city into six new rows, dividing them by race. Consequently, we adjusted the original population variable in the first dataset by multiplying the racial population proportion with the total population. This allowed us to obtain the proportion of medical cases attributed to each race in each city. In the modeling and prediction process, we will use merged dataset.

Additionally, we melted data by transforming the data from wide format to long format. The columns specified in id\_vars=['States', 'Ten-Year Age Groups'] remain as identifier variables, while the remaining columns specified in value\_vars are melted into a single 'Race' column and their corresponding values are stacked in a 'Population' column. The resulting DataFrame is assigned to melted\_data. The 'States' column becomes the index, the unique values in the 'NewCol' column become the new columns, and the corresponding 'Population' values are populated in the cells. The resulting DataFrame is assigned to pivot\_data. Finally, the reset\_index() method is called on the pivot\_data DataFrame to reset the index and convert the 'States' column back to a normal column. By performing these steps, we transformed the original data into a new format where each unique combination of 'States' becomes a row, and the 'Population' values are populated in the respective columns based on the combination of 'Ten-Year Age Groups' and 'Race'.

**Feature Engineering**

For building model, we need to convert categorical variables into dummy variables, which will be our predictors in prediction process. So, we decided to convert Year, Race and Category variables into dummy variables in merged dataset. Moreover, we observed that Geolocation variable can have a chance to shows exact point which contains longitude and latitude point. For making useful this column, we split Geolocation column into two columns under the Longitude and Latitude columns and converted their data types into float. When we have those variables, we won’t need to keep Geolocation, States and LocationName columns which are not beneficial for further analysis.

For applying DNN we uploaded torch, torch.nn packages on Python and converted our categorical variables (Category and Measure) into dummies. Then we converted the input features and target variable – ‘Data Value’ into PyTorch tensors and stores them. The \_\_len\_\_ method returns the length of the dataset, and the \_\_getitem\_\_ method allows indexing and retrieves a specific data sample from the dataset. The ‘MyModel’ class is a subclass of nn.Module, the base class for all neural network models in PyTorch. It defines the architecture of the model with three fully connected (linear) layers. The \_\_init\_\_ method initializes the layers, and the forward method defines the forward pass of the model.   
The modified ‘MyModel’ class includes additional layers and the nn.LeakyReLU() activation function. Into architecture of model, two additional fully connected layers (fc4 and fc5) have been added, each with different input and output sizes. The nn.LeakyReLU() activation function is applied after each fully connected layer to introduce non-linearity in the model. After this process, we configurated parameters by set up the mode, dataset, device, data loader, loss function, and optimizer for training the neural network mode. The device variable is set based on the availability of CUDA (GPU). If CUDA is available, the model and dataset are moved to the GPU using model.to(device) and dataset.to(device). The dataset is explicitly moved to the device using dataset = dataset.to(device). This ensures that the dataset is also on the GPU if available, which is required for seamless training. The device variable is assigned with 'cuda' in the line device = torch.device('cuda'). The learning rate for the optimizer is set to lr=0.0005.

**Predictive Analysis**

As a starting of predictive analysis, first, we defined our target variable – ‘Data Value’ which shows how many % of people have diseases. After encoding the categorical variables,

we divided our merged dataset into training and testing datasets within 80/20 proportion. Then we built Random Forest model with all features on training dataset and made predictions over the testing dataset. We measured Mean Absolute Error, Mean Squared Error, Root Mean Squared Error and R-squared. R-squared of random forest model is 98.09% which indicates that the predictors of model explain all the variance. Mean Absolute Error (MAE) measures the average absolute difference between the predicted and actual values. In this case, the average absolute difference is 1.96, which indicates that, on average, the predicted values are approximately 1.96 units away from the actual values.

Mean Squared Error (MSE) measures the average squared difference between the predicted and actual values. A lower MSE indicates better performance. In this case, the MSE is 12.15, which suggests that, on average, the squared difference between predicted and actual values is 12.15. Root Mean Square Error (RMSE) is the square root of the MSE and provides a measure of the average magnitude of the prediction error. In this case, the RMSE is 3.48, indicating that, on average, the predicted values are approximately 3.48, units away from the actual values. In a conclusion, these results indicate that the random forest model is performing well, with a high R2 score and relatively low errors.

Then we built a horizontal bar plot to visualize the feature importance, which helps us to understand which features contribute the most to the predictions.

A picture containing text, screenshot

Description automatically generated

Figure 15: Feature Importance of Random Forest with all variables

From Figure 15, we could observe that the most effective variables to output, are Category, Measure. Addition to this, Year, x, y, and TotalPopulation have impact on target variable. Unfortunately, we couldn’t see any effects of race over Data Value which shows percentage of people who have disease over this figure. So, we decided to remove important features to see how the race values are acting and got the following result.

As the second model, we built Random Forest excluding Measure, Category, and Year features. Then we assigned ‘Data Value’ as target and categorical variables were encoded, and the dataset was divided into training and testing datasets. We also made predictions on the test set using trained model, and calculated MAE, MSE, RMSE and R2 score. In this model, we got the average absolute difference is approximately 2.004, the MSE is approximately 12.38, the RMSE is approximately 3.52 and 98.06% in R2 Square. In summary, the model's performance can be considered relatively good based on these metrics. We realized that when excluding Measure, Category and Year features, our model are losing it performance in spite that increasing effects of races within different age groups.

A picture containing text, screenshot

Description automatically generated

Figure 16: Feature Importance of Random Forest excluding a few variables.

As seen in Figure 16, races have very little impact on ‘Data Value’ variable. Only these races within different age group – 25-34 years Asian, 65-74 years Black or African American and 15-24 years Asian are more contributors in people who get illness.

Additionally, we implemented AutoML model which checked all possible models and selected Random Forest Regressor as best model option which is great fact for our Random Forest model performance.

In the final, for selecting the best predictive models for our dataset, we implemented Deep Neural Network which contains multiple successive layers of neurons- inputs to get output layer. We started to trained loop for the neural network model which iterates over the dataset for a specified number of epochs and performs forward and backward passes, updating the model's parameters based on the calculated loss. The model consists of three fully connected layers with decreasing output sizes (128, 64, 1). The learning rate for the optimizer is set to lr=0.0005

The batch size was set to 128 and activation was 'nn.ReLU()’. Furthermore, we accumulated the running loss for the current batch and evaluated the average loss for the epoch and kept those epoch loss in the list. The loss value is a measure of how well the model is performing on the training data. It quantifies the discrepancy between the predicted output of the model and the actual output. A lower loss value indicates better performance, as it suggests that the model's predictions are closer to the true values. Our loss value in the epoch loss are fluctuating between 18-20 throughout training. We built a plot showing the training loss values over the range of epochs specified. This visualization can help in assessing the training progress and understanding how the loss evolves over time in Figure 17.

A picture containing text, screenshot, display, software

Description automatically generated

Figure 17: Comparison of Actual and Predicted Values in line graph

Moreover, we evaluated a trained model using a test dataset and then visualized the true values and predicted values and created plot which shows Actual and Predicted value:

A graph with blue and orange lines

Description automatically generated with low confidence

Figure 18: Comparison of Actual and Predicted Values in line graph

A picture containing screenshot, diagram, plot, graphics software

Description automatically generated

Figure 19: Comparison of Actual and Predicted Values in bar graph

From Figure 18 and 19, we observed that our predicted values in DNN model are significantly matching with actual values through the orange and blue lines. It means that the performance of DNN is very well.

Moreover, we calculated evaluation metrics; MSE and R-squared as 12.08 and 98.11%, respectively. A lower MSE indicated better performance, as it means the model's predictions are closer to the true values. A higher R-squared value suggests that the model captures a significant portion of the variability in the target variable. In a conclusion, these metrics show the model performs well, with relatively low MSE and a high R-squared value.

For reaching maximum performance, we built complex neural network model with long training and only using main variables – Category, Measure, Longitude and Latitude’ we increased model layers to 5 while decreasing output sizes (256,128, 64,32,1). In this model, the batch size was set to 128 and activation and learning rate kept same. In this model, we almost acquired most accurate level over the training loss and well matching between predicted and actual values which is clearly observed in Figure 20. In long trained Neural Network model, we gained lower MSE as 11.006 and highest R-squared as 98.28%. From this model result, we can conclude that having disease over the USA, different disease category, their measures and location are main factors.

A picture containing screenshot, diagram, plot, graphics software

Description automatically generated

Figure 20: Distribution of True and Predicted Values in bar graph

**Conclusion**

In our project, we aimed to gain meaningful insights from disease data and develop a predictive model to estimate disease prevalence. We utilized various models and techniques to achieve this objective. Based on these results, we could determine that our DNN model outperformed the Random Forest models in terms of predictive accuracy, achieving a lower MSE and higher R-squared value. This indicates that the DNN model captures the relationships between the variables more effectively, resulting in better predictions. From this model’s result and different analysis, we defined main factors to have diseases over the United States are Category, Measure, and location of disease. Considering those, decision-makers can make informed choices to address public health challenges effectively.

The insights gained from our project and the predictive models developed can have significant implications for public health interventions, resource allocation, and policy development. By understanding the relationships between variables and accurately estimating disease prevalence, decision-makers can make informed choices to address public health challenges effectively.

It is important to note that the accuracy and performance of our models are based on the available data and the specific features used. Further research, data refinement, and model optimization may be necessary to improve the accuracy and generalizability of the predictive models in real-world scenarios.

**Reference**

1. Centers for Disease Control and Prevention. (n.d.-a). *Places: Local data for Better Health, county data 2022 release*. Centers for Disease Control and Prevention. <https://chronicdata.cdc.gov/500-Cities-Places/PLACES-Local-Data-for-Better-Health-County-Data-20/swc5-untb>
2. CDC wonder. (n.d.-b). <https://wonder.cdc.gov/controller/datarequest/D176>
3. *Pandas.get\_dummies#*. pandas.get\_dummies - pandas 2.0.2 documentation. (n.d.). <https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html>
4. *Sklearn.ensemble.randomforestclassifier*. scikit. (n.d.). <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
5. Team, D. (2018, September 28). *Deep learning with python tutorial for Beginners - DNN & Ann*. DataFlair. [https://data-flair.training/blogs/deep-learning-with-python-tutorial/#](https://data-flair.training/blogs/deep-learning-with-python-tutorial/)
6. Real Python. (2021, February 27). *Python ai: How to build A neural network & make predictions*. Real Python. <https://realpython.com/python-ai-neural-network/>
7. *Deep Neural Network*. Deep Neural Network - an overview | ScienceDirect Topics. (n.d.). [https://www.sciencedirect.com/topics/computer-science/deep-neural-network#:~:text=Deep%20neural%20networks%20(DNN)%20is,et%20al.%2C%202016](https://www.sciencedirect.com/topics/computer-science/deep-neural-network)).