Montgomery Crash Data Visualization Tool

Computer Science Capstone

By: Sravani Yerramaneni

Western Governor's University

TABLE OF CONTENTS

LETTER OF TRANSMITTAL

Dear Mr. Elrich,

I'm excited for this opportunity to aid the Montgomery County Department of Transportation in the Vision Zero plan to eliminate all traffic deaths in Montgomery County by 2030. It is an ambitious goal and one that I am more than eager to take part in.

Road accidents are a major contributor to property damage, injuries, and deaths and yet roads play an unavoidable role in people's lives. Montgomery County's roads are no different.

After seeing a peak in road accidents in 2015, Montgomery County has made many strides towards maintaining safer roads by having the Montgomery County Council vote unanimously in 2015 to incorporate the Vision Zero plan to use data to eliminate road deaths and making the county's road accident datasets publicly available. The progress is noticeable and can be seen in the decrease of annual road accidents since 2015.

The proposed application is meant to make use of the now publicly available vehicle accident dataset and allow it be visualized in a format that can be easily understood and able to derive conclusions from. The product will serve as a publicly accessible web application and will feature a range of visualizations including a heat-map, a map of accident hotspots, and several more. The application will also support forecasting of the number of monthly accidents for each road.

While the application will not be the sole solution for the Vision Zero plan, it will serve as an important stepping stone in understanding the current condition of Montgomery County's roads. The main benefits of this application will be focused on two main areas: identification of the problems and a metric of success for any implemented solutions.

The total funding for this project will be $2,000 and consists mainly of the 2 weeks of hourly pay that compensates the developer's efforts. The developer responsible for developing this product for Montgomery County will be me. I have worked as a Data Scientist for the past 5 years now and I am skilled in the necessary languages and libraries needed to build and deploy the proposed product.

I am delighted to be able to aid the county in furthering its success regarding eliminating road deaths and I am looking forward to working together.

Sincerely,
Sravani Yerramaneni

PROJECT PROPOSAL

PROBLEM SUMMARY

The number of vehicle accidents and road deaths in Montgomery County peaked in 2015. Montgomery County has since launched a ‘Vision Zero’ action plan, an initiative to eliminate traffic deaths by 2030 in their county by rebuilding streets and intersections to be safer (Iannelli, 2017). The proposed application is to serve as a supplementary tool in these efforts. The product will be a web application featuring various interactive visualizations including a map revealing accident hotspots, a line graph of monthly accident counts per road along with a forecasting component, a pie chart showing the top roads with the highest accident counts, as well as a couple other similar visualizations.

APPLICATION BENEFITS

The application will serve as a visualization tool for Montgomery County’s DOT and citizens alike as a tool that transforms the publicly available road accident dataset into a format that can be used to easily draw conclusions from. The application will help citizens stay informed on the current state of their county’s roads and the application will aid MCDOT in their efforts to increase road safety in two main ways: identification of the afflicted road segments as well as a metric to measure the effectiveness of any future implementations to curb road accidents. By identifying the road segments prone to accidents through the map of accident hotspots, steps can be taken to either mitigate or resolve the issues along that area. After a solution is implemented, the number of accidents on the target road can be monitored and compared to the forecasted number of accidents as a metric for effectiveness.

APPLICATION DESCRIPTION

The application will be written in Python with the user interface customized using the ipywidgets library of Python. The Python application files along with the vehicle accident dataset in CSV format will be stored and maintained in a Github repository. Those files will then be deployed on Voila to form a web application.

DATA DESCRIPTION

The data is collected from the Montgomery County Police Department. It's publicly available in CSV format and contains 99,275 accident records spanning from January 2015 to September 2019. Although the dataset contains many columns, which include weather condition, cross-roads, municipality, speed limit, etc., for the purposes of this project, the only columns that will be used include:

- Road Name
- Crash Date/Time
- Injury Severity
- Longitude
- Latitude

Of these columns, only 'Road Name' contains null values, with 9057 null values, approximately 10% of the dataset. One thing to keep in mind is that an accident record is created for each vehicle involved in a crash. This means that for one accident, there could exist multiple records for each vehicle involved. When plotting accident count, a decision was made not to combine these records so that there will exist only one record per accident, because it was agreed upon that accidents involving more vehicles should hold greater weight than accidents involving less vehicles.

OBJECTIVE & HYPOTHESES

The main objective for this project is to contribute to a decrease in accident count in Montgomery County. For this to be achieved, the current state of the county's roads must be known and this project will greatly aid in doing that. It is hypothesized that if the application shows some road sections as places of frequent road accidents, county officials will consider improving or making changes to those road sections and the effects of these changes will be reflected back in the application in the form of reduced accident counts.

METHODOLOGY

The methodology used for this project will follow the Agile philosophy. Although there is a general idea of the requirements, the tools used to build the application are new to the developer and features of the application may change based on the tools' and developer's capabilities.

The product will be split into different components and built in iterations using the following steps:

- Define Requirements
    - Depending on the component, a general idea of the features and user interface will be formed.
- Design
    - The tools that will be used to implement the component are decided.

- Develop
    - The component is built.
- Test
    - The built component is tested to see if it meets requirements.
- Release
    - The built component is then shown to the client for approval. If changes are advised, there will be another, smaller iteration for that component. Once the component is approved, it will be deployed to Voila and the next iteration for a new component will start.

Since the product will be continuously deployed as it is developed, the final deployment into the production environment will be smooth and will mainly only require acceptance testing.

FUNDING REQUIREMENTS

Since the development environment and hosting site, Jupyter Notebook and Voila, respectively, are free, the only costs for this product will be the pay for the developer and minimal costs towards maintenance.

At $40/hour for 80 hours, the application development will cost $3200.

Maintenance will cost $400/year for about 20 hours/year at $20/hr.

STAKEHOLDERS IMPACT

The proposed product will impact the Montgomery County DOT by giving them a tool that allows them to see the current condition of the county’s roads in terms of road accidents. Using this tool will give them insight into what road sections are prone to accidents, what times accidents occur on certain roads, and allow them to effectively strategize accident prevention measures.

DATA PRECAUTIONS

There is no sensitive data as there is no personally identifying information and the full dataset is publicly available on the Montgomery County Maryland website at: https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Drivers-Data/mmzv-x632

DEVELOPER'S EXPERTISE

The developer has 5 years of experience working as a data scientist and has a Bachelor's in Computer Science. The developer is skilled in relevant Python libraries such as pandas, numpy, and matplotlib.

TECHNICAL PROJECT PROPOSAL

PROBLEM STATEMENT

Montgomery County has been making its road accident data publicly available for the past 5 years now in an effort to promote transparency and show its commitment towards minimizing road accidents. However, no tool currently exists to effectively visualize and analyze the road accident data collected by Montgomery County. Only when the data is transformed into an appropriate format, can conclusions be formed about the data and the dataset be utilized.

CUSTOMER SUMMARY

The main customers of this product are the Montgomery County Department of Transportation. The Montgomery County DOT is invested in maintaining safer roads and reducing the property damage, injuries, and loss of life caused by road accidents.

The proposed product will allow the MCDOT to effectively visualize the condition of the county's roads over time. The product can be used to identify road sections prone to accidents, identify the times that accidents occur on certain roads, and monitor accident count on roads over time along with a 3-month forecast to use as a metric when evaluating the effectiveness of accident prevention strategies.

Another important aspect of the product is that it is made publicly available and so the public is another customer of this product. Citizens of Montgomery County can use it to assess risk and stay educated on the current state of the county's vehicle accidents. Having the data visualizations made publicly available may also encourage others to create solutions for the exposed problems.

EXISTING SYSTEM ANALYSIS

Currently, the closest application Montgomery County has to a data visualization tool is a public website that contains a point map of fatal accidents, and two heat-maps of pedestrian and cyclist-involved crashes. While those visualizations allow the data to be visualized in a visual format, conclusions can not be easily drawn from them.

The proposed application will provide more of a focus towards identifying problem areas over time and will exceed the existing application in functionality with the addition of several new visualizations, namely: a pie chart showing the top 15 roads with the highest accident counts, a map displaying the locations of accident clusters of a certain density, and a heat-map with a time component that allows the user to see where accidents occurred within a chosen time range. The proposed application will also feature the charting of a time series showing monthly accident count for a certain road along with a 3-month forecast that may be used as a metric when evaluating the effectiveness of the MCDOT's accident prevention strategies.

DATA

The data is collected via a crash reporting system and reported by police in several counties in Maryland.

The dataset currently has 99,275 accident records and is available publicly as a CSV file on the following website:

- https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Drivers-Data/mmzv-x632

Although there are 43 columns in the dataset, only the following columns of the dataset are used:

- Road Name
- Crash Date/Time
- Injury Severity
- Longitude
- Latitude

Of these columns, only ‘Road Name’ contains null values, with 9057 null values, comprising approximately 10% of the dataset. However, the null values won’t impact the project much. The features of the application that will use the ‘Road Name’ column include the pie chart of the top roads with the highest accident counts, the plotting of monthly accident counts for each road, and the total number of accidents hourly for each road.

For all of these visualizations, only the top 50 roads with the highest accident counts will be utilized. There are 2796 unique road names in the dataset and it would be impractical to incorporate all road names into the application, especially considering that more than 80% of those roads have had less than 10 accidents over a 4 year span.

The only data column to manipulate would involve converting the ‘Crash Date/Time’ column to a datetime object in Python so the values can be treated accordingly.

The only maintenance that would have to be done would be to change the .CSV file routinely to reflect the new records added to the crash reporting system. The underlying program will be able to clean the data automatically without maintenance.

PROJECT METHODOLOGY

The application has a relatively small scope with only 5 main visualizations. The visualizations will for the most part, work individually with little to no integration required between them. The functional and visual elements of the application are not yet fully decided upon and so the tools that will be used to build the application are not completely decided upon either. For these reasons, the Agile philosophy will work best for this project. Using this methodology, the product will be split into components and built iteratively, allowing the client to have each component in their hands as soon as it is built.

Once the general requirements for the overall application are established with the client, each component of the project will be built iteratively following the following phases:

- Define Requirements
    - In this phase, the component of the application to be built in this iteration will be decided and a general idea of its requirements will be formed.
- Design
    - A user interface for the component will be sketched and taken for approval by the client. The tools that can be used to fulfill the task are considered and decided upon.
- Develop
    - The component will be built using the tools decided above. If needed, tools are changed according to their capabilities and relevance. Some features of the component may differ from the original plan based on the developer's and tools' capabilities.
- Test
    - Unit testing will be done in the development phase as and when the developer deems it necessary.
    - Functional testing will be done in and after the development phase to ensure that the correct information is being displayed in the graphs or figures.
    - Ad-hoc testing will take place once each component is built to ensure that the widgets work properly without breaking after some unexpected usage.

    - Integration testing will be done when integrating the new module of code with the previously tested modules. This will likely be minimal as the modules of code will interact very little.
    - Once the module completes its testing for the allotted time, acceptance testing will take place with the client in which the module completed in this iteration will be shown to the client for approval.
- Feedback
    - Once the acceptance testing with the client takes place, feedback will be taken for further improvements and the application will be revised as seen fit by revisiting the previous phases as necessary.
- Release
    - As each component is built and approved, the code files in the Github repository will be updated as well so the code can be deployed to Voila to be accessible to all end users. This also allows any errors that may arise when deploying to be found early in the process and not at the end of the project.

PROJECT OUTCOMES

The project outcomes are split into project and product deliverables.

- Project Deliverables
    - Milestones Schedule - This will be a schedule detailing the anticipated dates for each milestone as a measure of on-track progress
    - Test Plans - This is a document that will outline all the testing procedures that will take place to ensure the product meets quality standards

    - Wireframe - This will be delivered early in the project as a skeleton guide of the website that will be produced
    - Mockups - This will be delivered to the client as each component is being built to give an idea of how the finished component will look
    - User guide - This will be a very simple guide on how to access and utilize the application.
- Product Deliverables
    - Web Application - This will be a publicly accessible web application featuring 5 visualizations with the incorporation of interactive elements. The visualizations include:
        - Pie Chart - This will show the top 13 roads with the highest accident counts.
        - Line Graph of Time Series w/ Forecasting Feature - This will plot the number of monthly accidents for a chosen road within a chosen date range. A 3-month forecast of accident counts will also be provided.
        - Stacked Bar Graph - This will show the number of accidents a chosen road has had hourly. Each bar is divided into 5 sections divided by injury severity.
        - Map of Accident Hotspots - This will involve a clustering algorithm called DBSCAN that finds where accidents are densely populated according to the decided parameters. The clusters are then overlaid onto a map of the county roads.

        - Heatmap of Accidents over Time - This is a heat-map that shows the number of accidents that have happened in an area, displayed using a color scale to emphasize areas with more accidents over a chosen date range.
    - Source Code - All the code used for the proposed product will be stored in a github repository that the client has access to.
        - Machine Learning Models - The time series forecasting model known as SARIMA and the clustering algorithm known as DBSCAN will be included in the code along with their chosen parameters.

IMPLEMENTATION PLAN

Strategy

Since the MCDOT does not have any separate data visualization applications apart from the point/heat maps hosted on one of its websites mentioned above under ‘Existing System Analysis', the proposed product will be hosted on the web using Voila and there will be little in the way of existing systems to integrate apart from possibly linking to the proposed application’s website from MCDOT’s current website.

Phases of rollout

As the project is being built, each component will be viewed through the Jupyter Notebook Extension ‘Voila’. This will allow quick visualization of the product as a web application without deployment. As each component of the product is built, it will be viewed in Voila to ensure that any figures or interactive widgets appear and function properly.

Since the proposed product consists of several different modules that interact very little with each other, each visualization will be added to the Github repository as it is completed. This will make the process of deployment onto Voila very smooth as any issues that may arise in deployment will be caught in the early stages of the project rather than in the final stage. This also allows the client to have access to each component of the project as it is completed.

Once the application is completed and the MCDOT integrates the usage of the proposed application into their activities, feedback will be taken for further improvements and the application will be revised as seen fit.

Testing & Final Distribution

Ad-hoc testing will take place once each component is built to ensure that the widgets work properly without breaking after any unexpected usage.

Integration testing will be done when integrating the new module of code with the previously tested modules. This will likely be minimal as the modules of code will interact very little.

Once a module completes its allotted time for testing, acceptance testing will take place with the client in which the module will be shown in Voila to the client for validation. Once the component is approved, it will be added to the Github repository.

Milestones

The schedule for the milestones are detailed below in the section titled ‘Milestones’ and it will provide a flexible project schedule that can be used to measure whether progress is on-track.

Deliverables

- A web application that contains 5 visualizations with interactive elements.
- A simple user guide that details how to access and utilize the application.

User Testing

Once the MCDOT integrates the usage of the proposed application into their activities, feedback will be taken for further improvements and the application will be revised as seen fit.

EVALUATION PLAN

Testing will take place throughout the development of the application to ensure that the application meets functionality requirements including having the correct information being displayed on charts and figures. Once the developed application is complete and is integrated into the MCDOT's activities, feedback will be taken on further improving the application and will be done as seen fit.

The machine learning algorithms in this project, DBSCAN and SARIMA, were used for creating the clusters for the accident heat-map and forecasting the monthly accident count, respectively.

The parameters of the DBSCAN algorithm were chosen to show areas where at least 100 accidents occurred within a 40 meter radius of each other. These parameters were chosen to reduce the number of clusters shown down to 34 so as to make the map of accident hotspots more easily viewable. However, these parameters may be easily adjusted later based on need.

There are many metrics for evaluating the accuracy of a SARIMA model. A few are mean error, root mean squared error, and mean absolute error (Hyndman & Athanasopoulos, 2018). Two main methods for evaluating the model accuracy will be used. One is the root mean squared error. The ideal value for the root mean squared error will be determined on a case by case basis for each road. Another method that will be used for this project's purposes is to have the model plotted alongside the data, so that the deviations between each can be visually seen.

RESOURCES & COSTS

As the programming environment and hosting service are free, the only costs associated with this project involve the pay for the developer and the cost for yearly maintenance.

TYPE OF RESOURCE	DETAILS	COST
Software	● Anaconda ● Python 3.8.5 ○ Pandas ○ Numpy ○ Ipywidgets ○ Matplotlib ○ Bqplot ○ Folium	Free
Hardware	Provided by the developer: ● Computer with internet access and access to the required software	Free

Programming Environments	● Jupyter Notebooks ● Voila	Free
Hosting	● Github repository ● Voila	Free
Human Resources	There will only be one developer.	Development: ● Development: $40/hr for 80 hours = $3200 ● Maintenance: $20/hr for about 20 hours/year = $400/yr
TOTAL		$3200 + $400/year

TIMELINE & MILESTONES

For each component to be considered ‘built’, it must have undergone the required testing outlined in the ‘PROJECT METHODOLOGY’ section above.

MILESTONE	START	END	DURATION
Requirements are collected	Dec 31	Jan 1	10 hours
Time series forecasting model is built with minimal residuals	Jan 2	Jan 5	20 hours
Incorporate interactive elements for time	Jan 8	Jan 8	5 hours

series			
Clustering algorithm is built with the parameters adjusted appropriately	Jan 9	Jan 11	15 hours
Heat-map is built and deployed	Jan 12	Jan 13	10 hours
Stacked Bar Chart is built and deployed	Jan 14	Jan 15	10 hours
Pie Chart is built and deployed	Jan 16	Jan 16	5 hours
Final acceptance testing and hand-over	Jan 17	Jan 17	5 hours
COMPLETION DATE/ TOTAL HOURS		Jan 17	80 hours

POST-IMPLEMENTATION REPORT

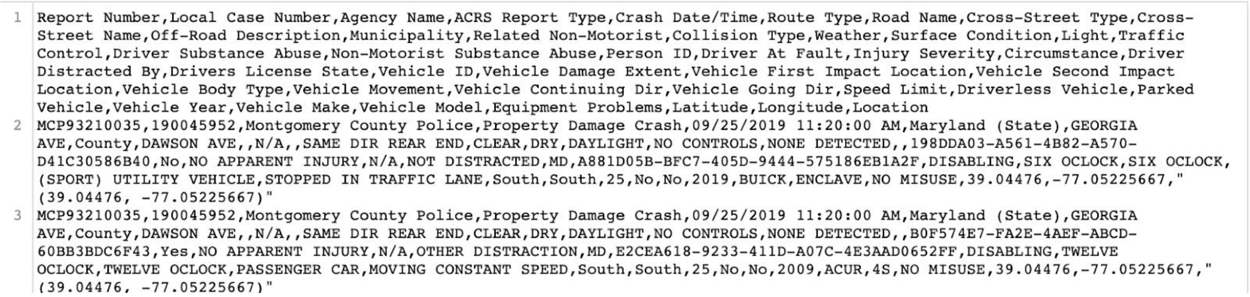
PROJECT PURPOSE

Montgomery County allotted a lot of funding for the Vision Zero plan, an ambitious project whose goal is to reduce traffic accident deaths in Montgomery County to 0 by 2030. The plan will involve rebuilding and restructuring several roads and intersections. Although the MCDOT had road accident data from 2015 to 2019, they didn’t have a tool they could use to visualize that data.

The original problem the application was meant to solve was providing the MCDOT with a data visualization tool that could easily be incorporated into accident prevention planning and be used to provide useful insight into the county's road conditions regarding traffic accidents. The product would achieve this by providing 5 main visualizations which would allow users to see the prevalence of accidents on each road, the locations of accidents, and the times in which accidents occur on each road.

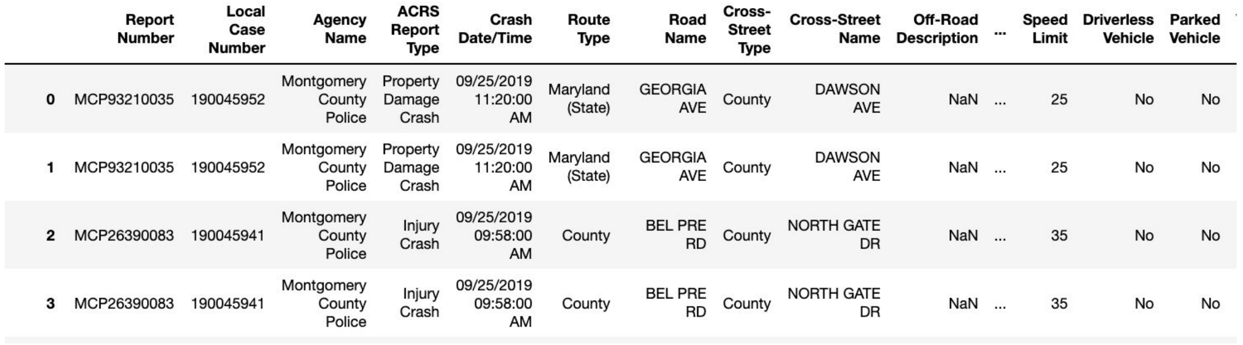
DATASETS

The raw dataset was in CSV format and contained 43 columns and 99,275 accident records spanning 2015 to 2019.



```
Report Number,Local Case Number,Agency Name,ACRS Report Type,Crash Date/Time,Route Type,Road Name,Cross-Street Type,Cross-
Street Name,Off-Road Description,Municipality,Related Non-Motorist,Collision Type,Weather,Surface Condition,Light,Traffic
Control,Driver Substance Abuse,Non-Motorist Substance Abuse,Person ID,Driver At Fault,Injury Severity,Circumstance,Driver
Distracted By,Drivers License State,Vehicle ID,Vehicle Damage Extent,Vehicle First Impact Location,Vehicle Second Impact
Location,Vehicle Body Type,Vehicle Movement,Vehicle Continuing Dir,Vehicle Going Dir,Speed Limit,Driverless Vehicle,Parked
Vehicle,Vehicle Year,Vehicle Make,Vehicle Model,Equipment Problems,Latitude,Longitude,Location
MCP93210035,190045952,Montgomery County Police,Property Damage Crash,09/25/2019 11:20:00 AM,Maryland (State),GEORGIA
AVE,County,DAWSON AVE,,N/A,,SAME DIR REAR END,CLEAR,DRY,DAYLIGHT,NO CONTROLS,NONE DETECTED,,198DDA03-A561-4B82-A570-
D41C30586B40,No,NO APPARENT INJURY,N/A,NOT DISTRACTED,MD,A881D05B-BFC7-405D-9444-575186EB1A2F,DISABLING,SIX OCLOCK,SIX OCLOCK,
(SPORT) UTILITY VEHICLE,STOPPED IN TRAFFIC LANE,South,South,25,No,No,2019,BUICK,ENCLAVE,NO MISUSE,39.04476,-77.05225667,"
(39.04476, -77.05225667)"
MCP93210035,190045952,Montgomery County Police,Property Damage Crash,09/25/2019 11:20:00 AM,Maryland (State),GEORGIA
AVE,County,DAWSON AVE,,N/A,,SAME DIR REAR END,CLEAR,DRY,DAYLIGHT,NO CONTROLS,NONE DETECTED,,B0F574E7-FA2E-4AEF-ABCD-
60BB3BDC6F43,Yes,NO APPARENT INJURY,N/A,OTHER DISTRACTION,MD,E2CEA618-9233-411D-A07C-4E3AAD0652FF,DISABLING,TWELVE
OCLOCK,TWELVE OCLOCK,PASSENGER CAR,MOVING CONSTANT SPEED,South,South,25,No,No,2009,ACUR,4S,NO MISUSE,39.04476,-77.05225667,"
(39.04476, -77.05225667)"
```

In the program, the CSV file was converted into a pandas dataframe object. No cleaning of the dataset was necessary, however the 'Crash Date/Time' column was converted from strings into datetime objects so that the program could treat them as such.



	Report Number	Local Case Number	Agency Name	ACRS Report Type	Crash Date/Time	Route Type	Road Name	Cross-Street Type	Cross-Street Name	Off-Road Description	...	Speed Limit	Driverless Vehicle	Parked Vehicle
0	MCP93210035	190045952	Montgomery County Police	Property Damage Crash	09/25/2019 11:20:00 AM	Maryland (State)	GEORGIA AVE	County	DAWSON AVE	NaN	...	25	No	No
1	MCP93210035	190045952	Montgomery County Police	Property Damage Crash	09/25/2019 11:20:00 AM	Maryland (State)	GEORGIA AVE	County	DAWSON AVE	NaN	...	25	No	No
2	MCP26390083	190045941	Montgomery County Police	Injury Crash	09/25/2019 09:58:00 AM	County	BEL PRE RD	County	NORTH GATE DR	NaN	...	35	No	No
3	MCP26390083	190045941	Montgomery County Police	Injury Crash	09/25/2019 09:58:00 AM	County	BEL PRE RD	County	NORTH GATE DR	NaN	...	35	No	No

For each visualization, a new dataframe was created by extracting only the columns pertaining to the specific visualization from the main dataframe. For example, for the heatmap, only the ‘Datetime’, ‘Longitude’, and ‘Latitude’ columns were needed so a new dataframe called df_hm was created by extracting those columns from the main dataframe.

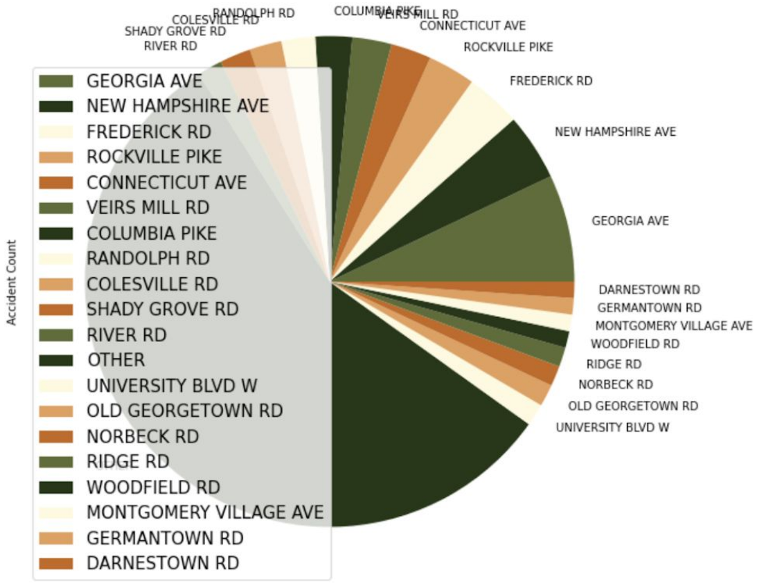
HYPOTHESIS VERIFICATION

To verify the hypothesis that changes to the roads will be made according to problems identified through the application, a simple questionnaire to the MCDOT will suffice. Another more concrete option could involve comparing the accident count from before and 2 years after the application was implemented.

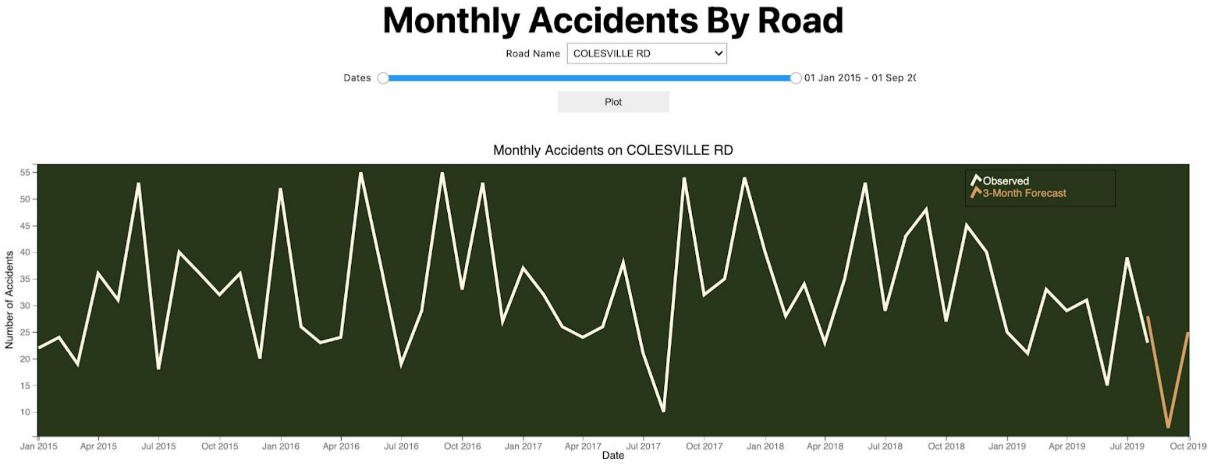
EFFECTIVE VISUALIZATIONS & REPORTING

Since there are only 5 main visualizations, they are all located vertically on the page, with each visualization centered and preceded by a title describing their purpose.

Top 20 Roads with the Highest Accident Counts



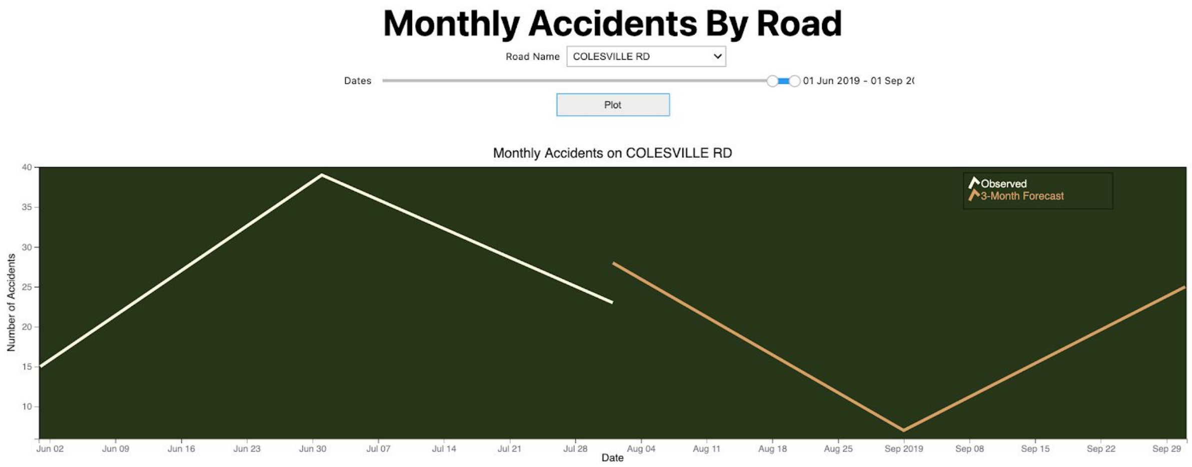
The first visualization is a pie chart showing the top 20 roads with the highest accident counts. The legend is large, covering more than half the circle and lists the top 20 roads. This data is found by finding how frequently each ‘Road Name’ appears in the dataset and then only extracting the value counts for the top 20 roads. The rest of the accident counts are summed and labeled as ‘OTHER’.


Monthly Accidents By Road

Road Name: COLESVILLE RD

Dates: 01 Jan 2015 - 01 Sep 2(

The second visualization is a line graph plotting the monthly accident counts for a specified road within a specified date range. This visualization features three interactive widgets, a dropdown showing the top 50 road names with the highest accidents, a selection range slider that allows the user to pick from the date range available in the dataset with monthly intervals, and a button to plot the specified options. The road names for the dropdown were identified using the same method as the one used for the pie chart. The ends of the date range were chosen to be the minimum and maximum dates of the dataset. A 3-month forecast is also provided using the SARIMA model and is distinguished on the graph by its color, identified on the legend.

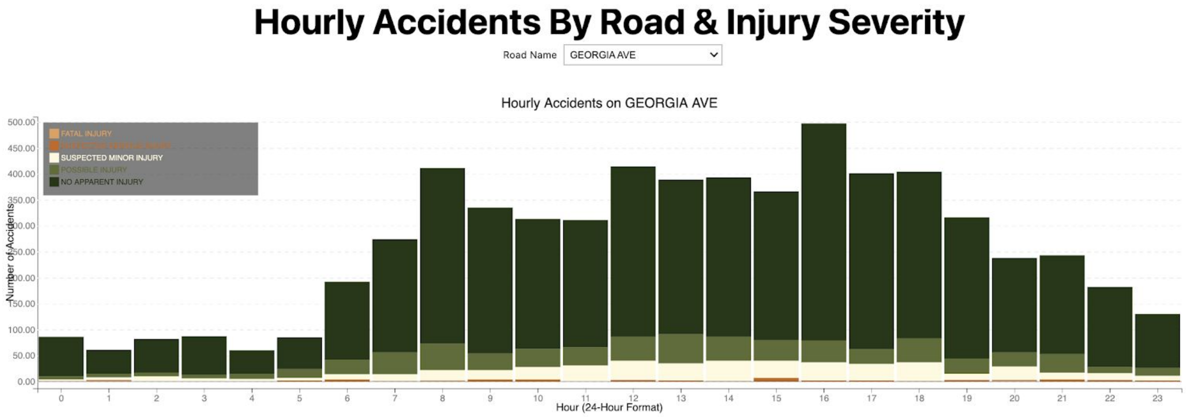


Monthly Accidents By Road

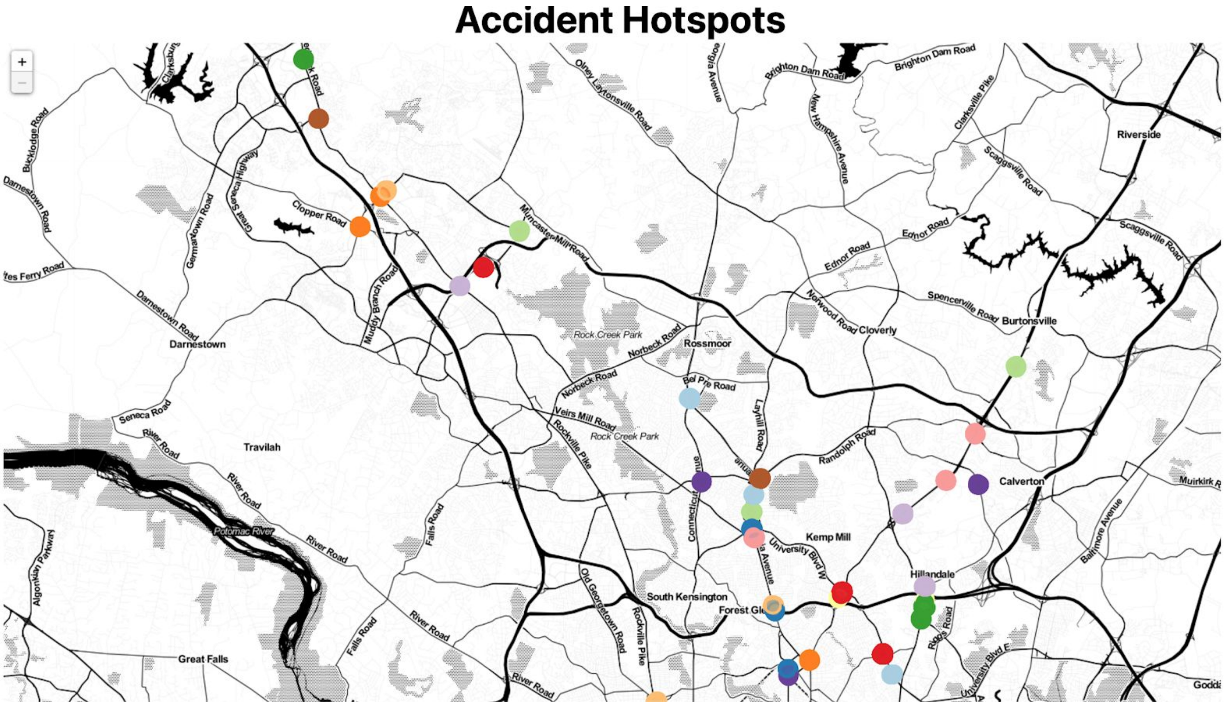
Road Name: COLESVILLE RD

Dates: 01 Jun 2019 - 01 Sep 20

The user can also zoom in on a certain time period by changing the date range as shown above. The ylim of the graph is set to change to accommodate the values on the graph.

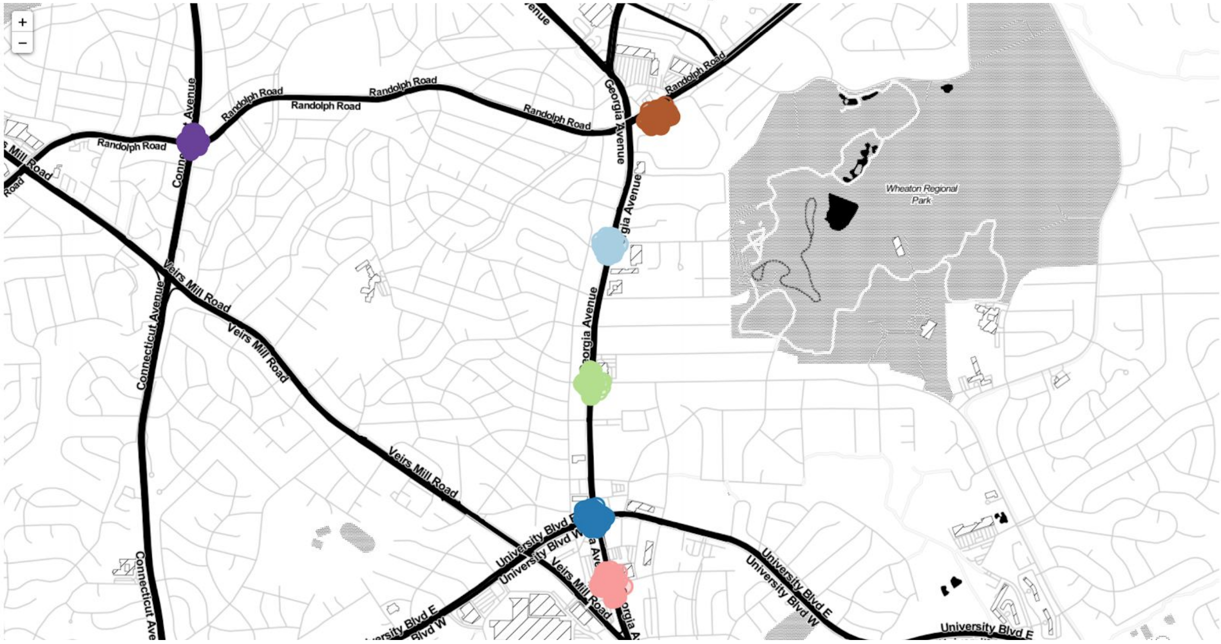


The third visualization is a stacked bar chart that shows the total number of accidents that occurred each hour for a specified road. The bar chart is stacked as it divides the accidents that occur into 5 categories based on 'Injury Severity'.



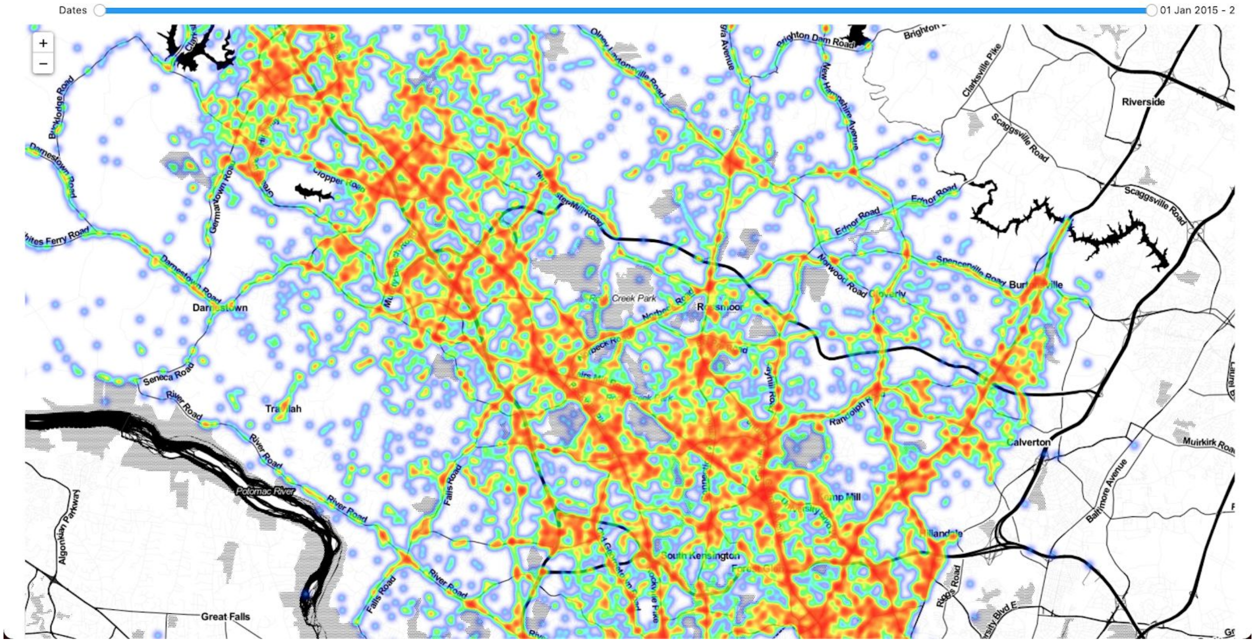
The fourth visualization is a map that shows accident hotspots. The tiles for the map were chosen by the ease with which the names of the roads can be identified. And the clusters were brightly colored so as to be easily distinguishable from the background map.

Accident Hotspots

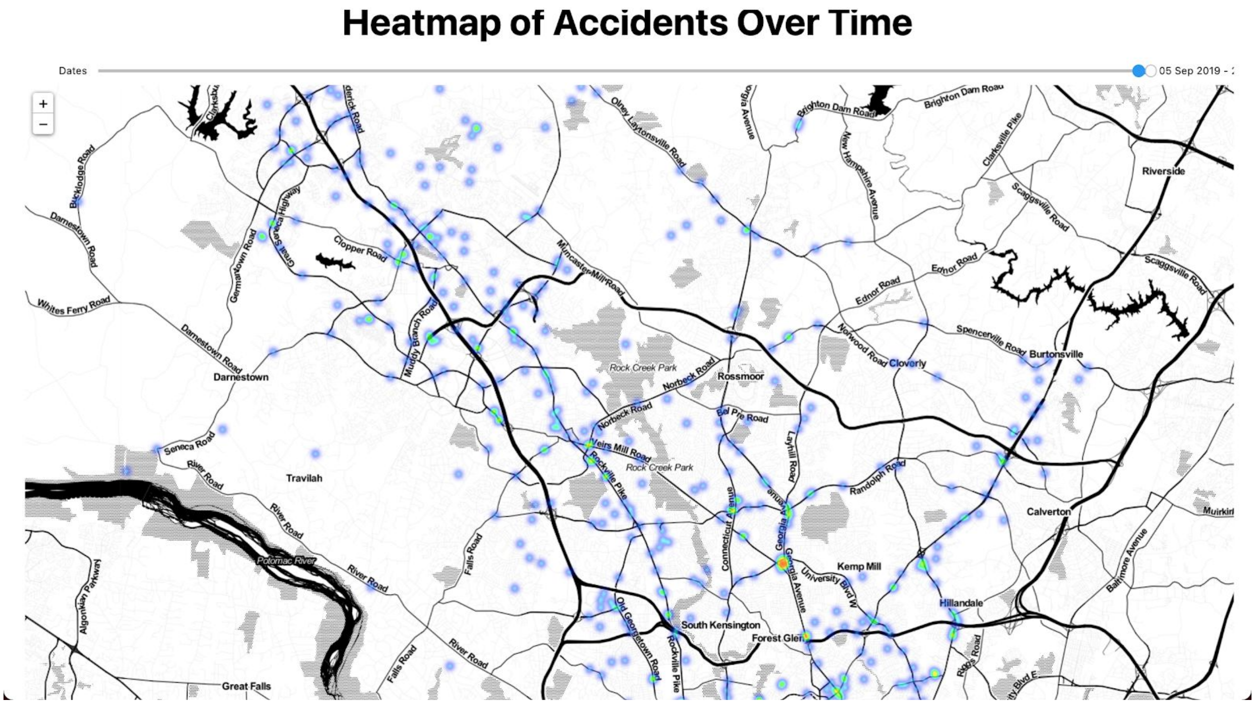


The map can also be zoomed in or out to see a more or less detailed view.

Heatmap of Accidents Over Time



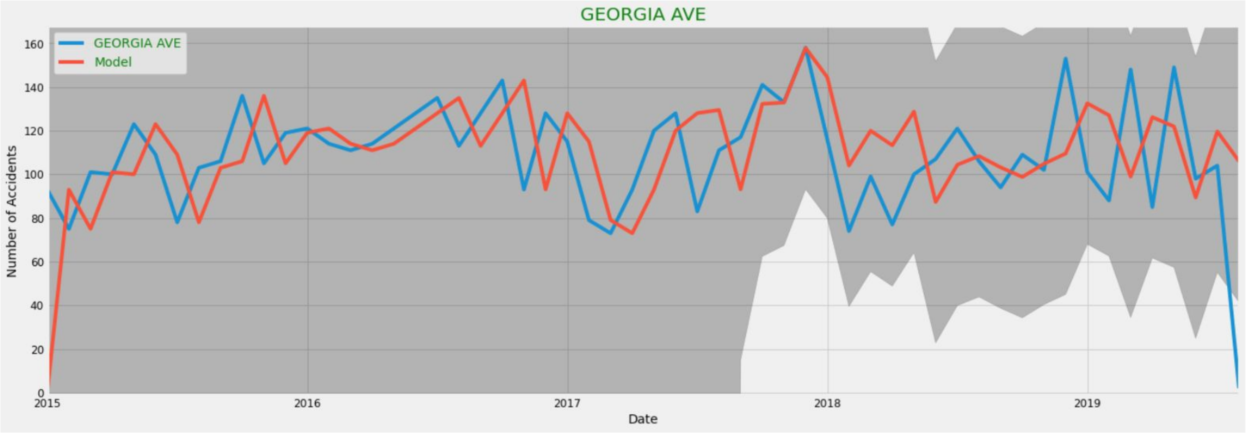
The fifth visualization is a heatmap that shows the frequency with which accidents occurred through a color scale over a specified time range. The same selection range slider used for the dates for the line chart is used here as well.



The selection range slider can be adjusted to see the accidents that occurred within smaller date intervals as well.

ACCURACY ANALYSIS

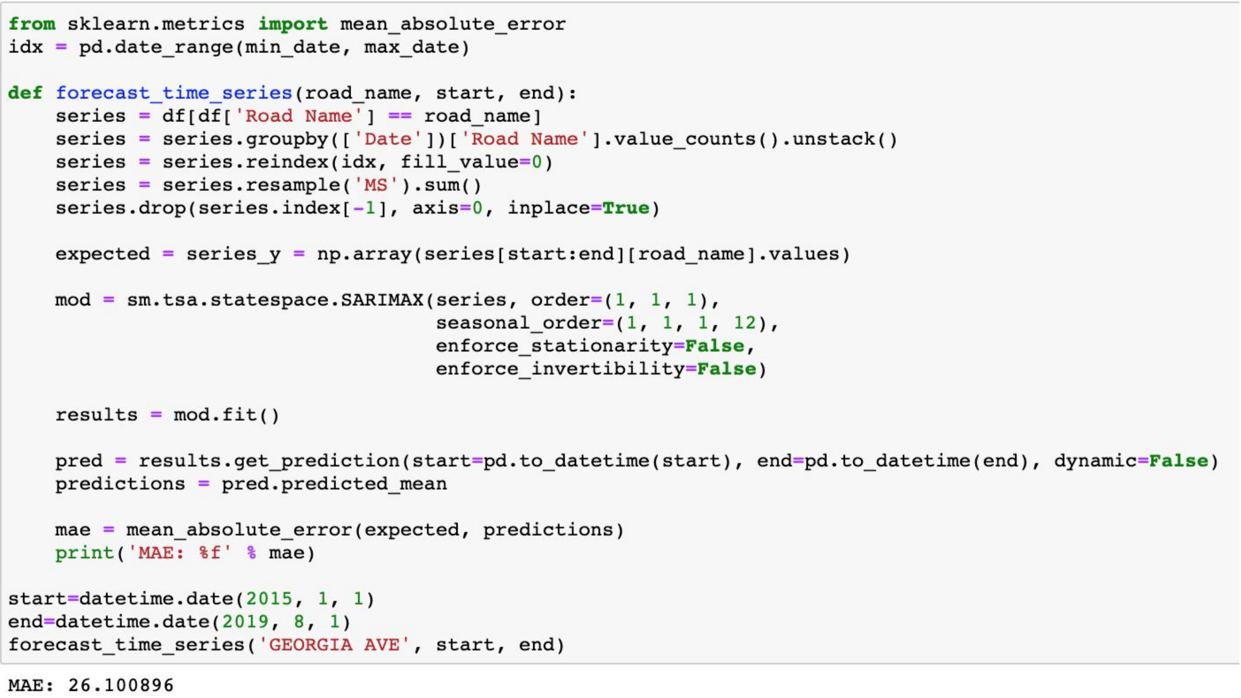
The machine learning algorithm used in this application was SARIMA and it was used to predict the monthly accident count of a road.



The chart above shows some of the preliminary predictions of the SARIMA model plotted against the observations.

The parameters for SARIMA were chosen by running the model using a grid search, which means testing all possible parameters and computing their AIC score. The parameters that yield the lowest AIC score are the best ones for the model.

The metric used to assess the accuracy of the model was Mean Absolute Error which takes the sum of the residuals (the value differences between the predicted and expected values) and divides the number of data points.



```
from sklearn.metrics import mean_absolute_error
idx = pd.date_range(min_date, max_date)

def forecast_time_series(road_name, start, end):
    series = df[df['Road Name'] == road_name]
    series = series.groupby(['Date'])['Road Name'].value_counts().unstack()
    series = series.reindex(idx, fill_value=0)
    series = series.resample('MS').sum()
    series.drop(series.index[-1], axis=0, inplace=True)

    expected = series_y = np.array(series[start:end][road_name].values)

    mod = sm.tsa.statespace.SARIMAX(series, order=(1, 1, 1),
                                    seasonal_order=(1, 1, 1, 12),
                                    enforce_stationarity=False,
                                    enforce_invertibility=False)

    results = mod.fit()

    pred = results.get_prediction(start=pd.to_datetime(start), end=pd.to_datetime(end), dynamic=False)
    predictions = pred.predicted_mean

    mae = mean_absolute_error(expected, predictions)
    print('MAE: %f' % mae)

start=datetime.date(2015, 1, 1)
end=datetime.date(2019, 8, 1)
forecast_time_series('GEORGIA AVE', start, end)
```

```
MAE: 26.100896
```

The resultant MAE was 26. Considering that the expected values ranged between around 80 and 150, which has a range of 70 values, the MAE wasn't too good. However, after the product is implemented and further feedback is taken regarding this feature, work can be done to improve this feature.

APPLICATION TESTING

- Functional Testing = Functional testing was done all throughout the project. As much of the project relies on visualizing the underlying dataset, every visualization had to first be conducted using a small sample of the data to ensure that the correct values were being displayed. There were also opportunities where the ordering of certain elements had to be synchronized with other elements to produce the correct plot. One example is with plotting, when the labels and values were manually declared, steps had to be taken to ensure that the labels and values were mapped correctly.

- Integration Testing = Although the modules interacted little to none at all, sometimes in the development environment, integrating the running module back into the code would introduce some errors. These errors mainly involved imports or variables not being available to the current cell because of the order in which the cells were run.

- Usability Testing = Usability testing was done at the end of each module to ensure that the product was intuitive to use. Although some minor flaws still remain, like the dates not being fully visible beside the date ranges and the pie chart labels overlapping, it was determined that the time and effort needed to fix those issues will likely introduce new, larger issues and so fixing those errors were given less priority.

APPLICATION FILES

To execute the product as a web application, Voila must be installed. Instructions and resources are provided in the 'USER'S GUIDE' section below.

The files will come in a .zip folder containing the following:

- written_capstone.pdf
- capstone_video.mp4
- A folder labeled 'code':
    - app.ipynb
    - crash.csv

USER’S GUIDE

To execute the product as a web application, Voila must be installed.

1. Install Voila. The instructions to do this are available on this website: https://voila.readthedocs.io/en/stable/install.html
2. Download the project files (capstone.ipynb and crash.csv) and save them in a known, accessible directory.
3. Open up your computer’s terminal and navigate to the directory where the project files are stored.
    a. This can be done using the ‘cd’ command. More information on this command is available here: https://linuxize.com/post/linux-cd-command/
4. Run the command ‘voila app.ipynb’
5. The application should open up in a web browser. It may take 1-2 minutes to load completely.

SUMMATION OF LEARNING EXPERIENCE

My prior experience with Jupyter Notebooks and the Python language assisted me greatly. I was able to solve the functional portion of the project involving manipulating the data frames and plotting the data relatively quickly. However, the user interface took a long time to understand and develop. Specifically, the ipywidgets and bqplot libraries were new to me and finding the relevant documentation for them proved to be very difficult, specifically for the bqplot library.

Another difficult portion of the project was deploying. Originally, the product was meant to be deployed as a web application which anyone can access with the URL. However, after many different options (Anvil, Heroku, Binder, etc.) were attempted and abandoned after many hours and days spent debugging, the product finally came to be deployed on Voila. While the application cannot be shared through a simple URL anymore, the process still remains relatively easy.

Although, I did not have direct access to any individuals with the knowledge and experience required to help me, many online forums such as stackoverflow, came in handy when attempting to solve any problem, although the less used the library, the harder it was to find relevant forums.

Overall, the process of customizing the webpage with the use of the ipywidgets and bqplot libraries was very rewarding, especially when looking back at the previous iterations of the project. However, it was also important to recognize the priority of a task and when the task should be abandoned accordingly, especially regarding the user interface. The most unpleasant experience involved the deployment as none of the tutorials attempted worked for my particular scenario. Having someone experienced with the tool (Heroku, Binder, etc.) would have greatly helped me as the provided documentations were quite difficult to follow with my limited level of knowledge in those areas.

Sources

Hyndman, R. J., & Athanasopoulos, G. (2018). Evaluating forecast accuracy. In *Forecasting: Principles and practice*. Heathmont, Vic.: OTexts.

Iannelli, N. (2017, November 03). Montgomery Co. aims for zero traffic deaths by 2030. Retrieved January 17, 2021, from https://wtop.com/montgomery-county/2017/11/montgomery-county-vision-zero/