Pritam shrestha DSC_680

May 29, 2021 Prof. Fadi Alsaleem

## Abstract

Many people are protesting and marching together after George Floyd's murder by the police force and some people are demanding reform and defunding the police force. So, it is one of the country's growing concerns because people are dividing in favor of him and against him. I am not on either side of them and will work as a neutral person. Furthermore, I will use machine learning algorithms to classify the death of people based on their race. My primary concern is classifying the race of victims based on age, gender, threat level, armed and mental illness, and finding the bias between white class people and black class people.

In addition, Initially, I will start this project implementing exploratory data analysis and will end implementing multiple machine learning algorithms to find the best model performance and parameters.

**Used resources:**

Resources are something that is used to complete the project. For example, here, I have used the following resources to complete this project.

1) Data source:
2) Other technical resources: Anaconda, Jupyter Notebook and python libraries,

## Data Source

Data is distinct pieces of information, usually formatted in a special way. It can be used to analyze and visualize implementing different machine learning algorithms. Data is the most valuable thing in data science because everything relies on it. Bad data could lead to bad output and project failure. So, data finding is a major milestone of the project. Finally, I found data and downloaded data as a CSV file format from the following source.

**Source: https://www.kaggle.com/amark720/police-killings**

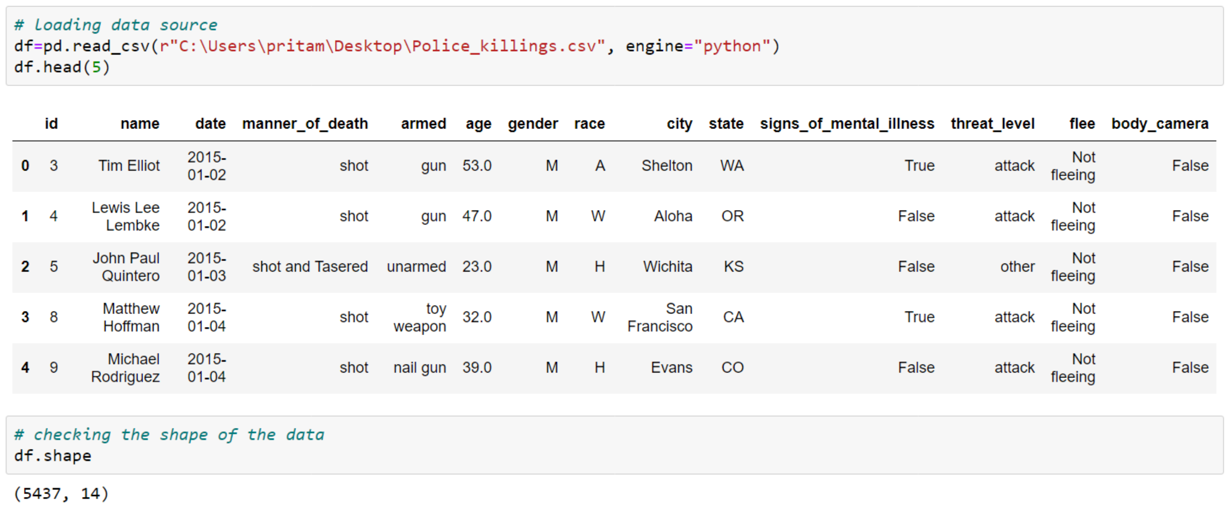
## Data understanding:

Understanding the problem statement is the best way of finding a solution. To understand the problem, we need to find the problem first. Problem finding is not easy because it could be related to anything. So first of all, we need to categories the type of problem, whether it is solvable or not. If it is solvable, then they need to pay attention to it. Likewise, I figured out the problem statement and started to find possible factors being responsible for prediction.

The structure of the data is provided as an image. Basically, I have downloaded the data from Kaggle and saved it in a CSV file. Later I read that data using the pandas library of python for further analysis.

## Structure of the data:

Before implementing the data for modeling, we have to analyze data to understand it better and perform exploratory data analysis. Raw data is not always ready for model implementation, so data preparation is another essential part of the data science project. Here is the structure of the downloaded data.



```
# Loading data source
df=pd.read_csv(r"C:\Users\pritam\Desktop\Police_killings.csv", engine="python")
df.head(5)
```

| | id | name | date | manner_of_death | armed | age | gender | race | city | state | signs_of_mental_illness | threat_level | flee | body_camera |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 3 | Tim Elliot | 2015-01-02 | shot | gun | 53.0 | M | A | Shelton | WA | True | attack | Not fleeing | False |
| **1** | 4 | Lewis Lee Lembke | 2015-01-02 | shot | gun | 47.0 | M | W | Aloha | OR | False | attack | Not fleeing | False |
| **2** | 5 | John Paul Quintero | 2015-01-03 | shot and Tasered | unarmed | 23.0 | M | H | Wichita | KS | False | other | Not fleeing | False |
| **3** | 8 | Matthew Hoffman | 2015-01-04 | shot | toy weapon | 32.0 | M | W | San Francisco | CA | True | attack | Not fleeing | False |
| **4** | 9 | Michael Rodriguez | 2015-01-04 | shot | nail gun | 39.0 | M | H | Evans | CO | False | attack | Not fleeing | False |

```
# checking the shape of the data
df.shape
```

```
(5437, 14)
```

Fig1: Structure of the datasets

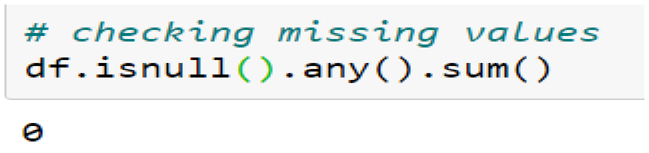
The datasets consist of 5437 rows and 14 columns. These datasets look straightforward. I will use these datasets for model implementation because my dataset is simple and small and won't generate any hardware problem or processing problem. However, before implementing the model, we should go through the Exploratory data analysis steps to avoid errors and issues.

## Exploratory data analysis:

Exploratory data analysis is an approach to analyzing the dataset by summarizing its main characteristics by implementing visual methods. The primary purpose of EDA is to see what the data can tell us beyond the formal modeling or hypothesis testing task. Unfortunately, data is not always ready for further analysis, so I did some exploratory data analysis using python.

1) Replaced header: I replaced the header of the downloaded data to make it more readable and understandable.
2) Fixed casing and inconsistent data: When I checked the data structure, I did not find any missing data. In contrast, I have found a few columns such as ID, City, Name, etc., which are

not correlated with the target variables. Similarly, some columns had many NaN values with the non-descriptive heading, so I have dropped off these columns to prepare the final dataset.

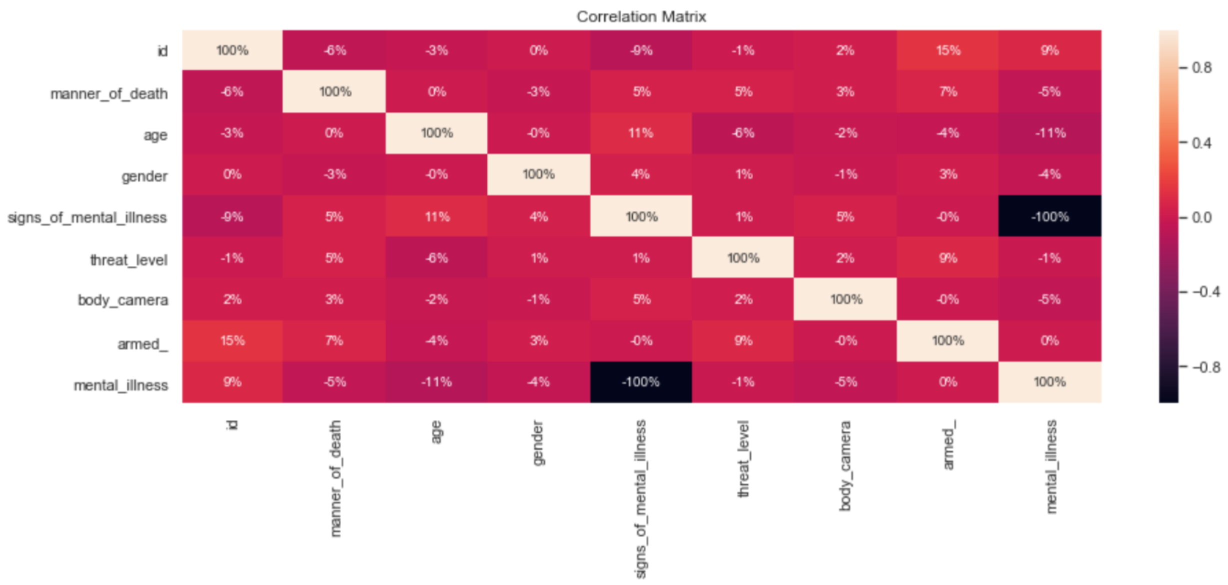


```
# checking missing values
df.isnull().any().sum()
```

```
0
```

Fig2: checking missing data

3) Checked outliers and bad data: Sometimes, we get outlier. Outlier means extreme values that fall a long way outside of the other observations. Finding outliers and proper handling of the outlier is very important; otherwise, it might lead to the wrong output or result and might bias to the higher value or majority dataset. For example, my dataset is a balanced dataset because it has 2485 white cases and 1299 black cases.
4) Data cleansing or cleaning: It is a process of finding and correcting corrupt or inaccurate records from the dataset. I was fortunate, so I have not found any corrupt data.
5) Data transformation: It is a process of converting data from one format to another format that makes data more readable and understandable. I had most many columns with categorical data so, I converted it into numerical data using the factorial method and finally used StandardScaler for feature scaling.
6) Visualization: It is a pictorial representation of data. It helps us analyze the data, so I plotted a couple of plots and found important information about people's death based on their race.



| | id | manner_of_death | age | gender | signs_of_mental_illness | threat_level | body_camera | armed_ | mental_illness |
|---|---|---|---|---|---|---|---|---|---|
| id | 100% | -6% | -3% | 0% | -9% | -1% | 2% | 15% | 9% |
| manner_of_death | -6% | 100% | 0% | -3% | 5% | 5% | 3% | 7% | -5% |
| age | -3% | 0% | 100% | -0% | 11% | -6% | -2% | -4% | -11% |
| gender | 0% | -3% | -0% | 100% | 4% | 1% | -1% | 3% | -4% |
| signs_of_mental_illness | -9% | 5% | 11% | 4% | 100% | 1% | 5% | -0% | -100% |
| threat_level | -1% | 5% | -6% | 1% | 1% | 100% | 2% | 9% | -1% |
| body_camera | 2% | 3% | -2% | -1% | 5% | 2% | 100% | -0% | -5% |
| armed_ | 15% | 7% | -4% | 3% | -0% | 9% | -0% | 100% | 0% |
| mental_illness | 9% | -5% | -11% | -4% | -100% | -1% | -5% | 0% | 100% |

Fig3: Correlation metrics

In the above figure, we can clearly see the correlation between the dataset's features that helps us figure out which features are responsible for the death. The red color indicates a strong(positive) relationship, and the black color indicates a weak( negative) relationship among the features.

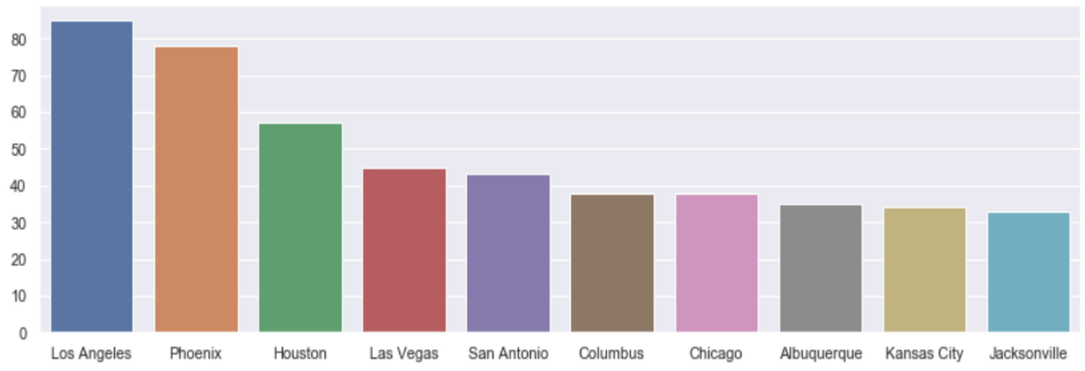


Fig4: 10 most dangerous city in the USA

The above bar plot clearly indicates that Los Angeles is the most dangerous city in the USA because it more than 80 death cases. Similarly, Jacksonville is the 9th dangerous city in the USA based on the death count.

**Feature Selection and Feature scaling:**

**Feature Selection:**

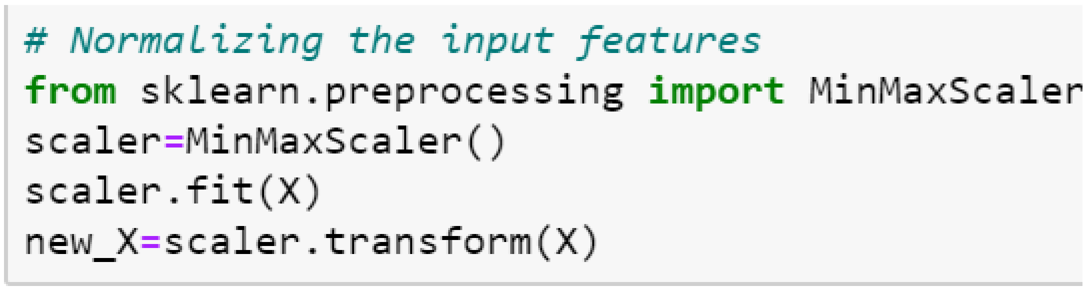
Feature selection is the process of selecting a subset of relevant features or variables for modeling or use in the model construction. It is also called variable selection. In my case, I have 14 variables or features; some of them are not correlated to the target variable, so I did not include them as my input variables for modeling. Instead, I only chose the following variables as my input variables and target variables.

1) Input features age, manner_of_death, gender, threat_level, armed_, and mental_illness.
2) Target variable: racewise_death

In the process of feature selection, I did not include some of them because they are not correlated to the target variable such as name, id, city, etc.

**Feature scaling:**

Feature scaling is the method to standardize the independent features present in the data. This is also called the normalization method. If it is not done, then machine learning algorithms tend to weight greater values higher and consider smaller values as the lower values, regardless of the unit of the values. So, to avoid this problem, we can use either normalization or standardization, but I have performed a normalization method to make my input features a normal and standard format.

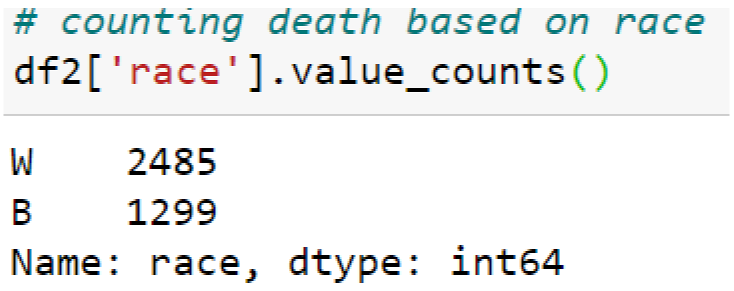


```
# Normalizing the input features
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(X)
new_X=scaler.transform(X)
```

Fig5: data normalization

**Handling imbalanced datasets:**

Dataset is the main asset of any data science project. An output directly depends on the nature of the datasets. Highly imbalanced data could lead to the wrong output because the model might be biased towards the majority of datasets. So, handling the imbalanced dataset is the best practice of data preparation in data science. It can be handled using different methods such as oversampling and downsampling. Here, I have not implemented this technique because my dataset is a balanced dataset.



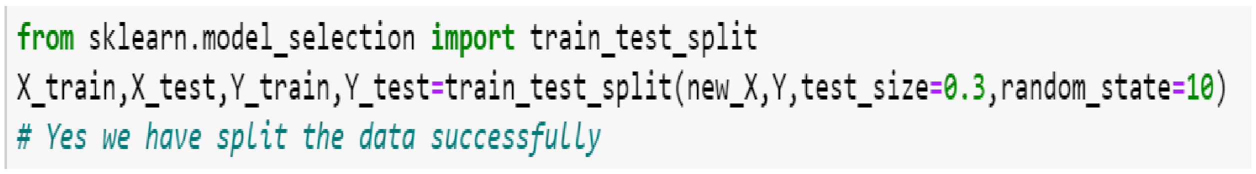
```
# counting death based on race
df2['race'].value_counts()
```

```
W     2485
B     1299
Name: race, dtype: int64
```

Fig6: target value counts

**Data Modeling:**

Data modeling means training machine learning algorithms to predict the target from the features. For the implementation of the model, we have to split out data into two data sets, such as training dataset and testing dataset, so I have split my input and target dataset into two datasets.



```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(new_X,Y,test_size=0.3,random_state=10)
# Yes we have split the data successfully
```

Fig7: data splitting using sklearn

1) Training dataset: It contains 70% of data for training purposes.

2) Testing dataset. It contains 30%  of data for testing purposes.

**Model selection:**

Now input and target data are ready to train the model and test the model, so I have selected two models for data modeling.

1) Logistic regression:

 Logistic regression is a supervised learning algorithm used to predict the probability of the target variable.

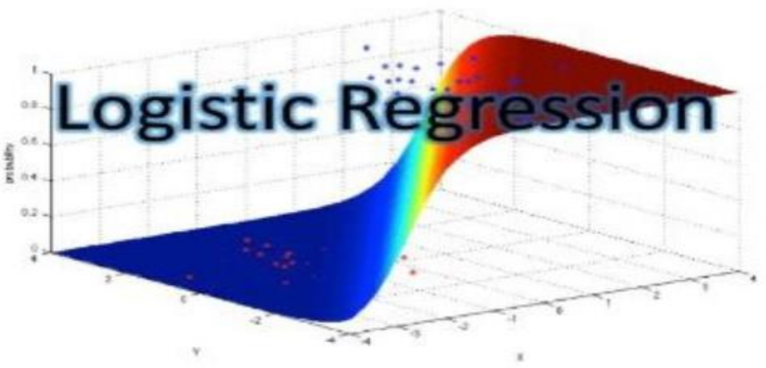


Fig8: Logistic regression

Source: https://medium.com/greyatom/logistic-regression-89e496433063

In my dataset, the dependent variable is the binary variable that contains data coded as 1 for black and 0 for white. It means it predicts P(Y=1) as a function of X.

I have considered a scenario where I need to classify whether a victim is white or black.

2) Random forest :

Random forest is an ensemble learning method for the classification problem; it performs tasks constructing a multilevel of the decision tree. In most cases, it gives a good result for classification problems. Generally, it works in the following ways.

1) First, it starts with the selection of random samples from a given dataset.

2) Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
3) In this step, voting will be performed for every predicted result.
4) At last, select the most voted prediction result as the final prediction result.

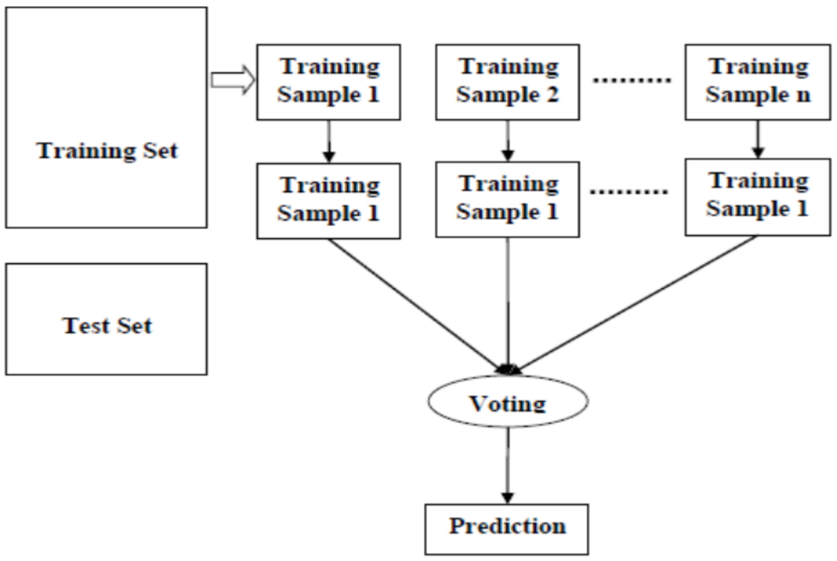


Fig9: Random forest

Hence, I have labeled data; it means I have both input data and target data, so it is called supervised learning algorithms and can be used to train the model with the supervision of the labeled data. Here I am not predicting anything; I am only classifying the target based on the available input features. So this type of problem is called classification problem, and I have only two classes, such as White and Black class. It is also called a binary classification problem.

**Hyperparameter Tuning:**

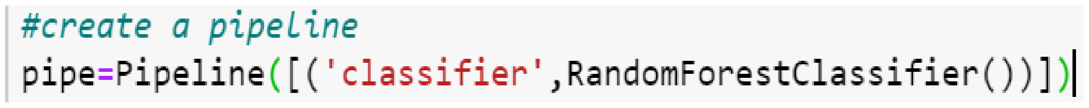
In machine learning, hyperparameter tuning is the problem of choosing a set of optimal hyperparameters for the learning algorithm. That helps us to give the best result. So I have selected different hyperparameters for regression modeling and random forest modeling.

**Pipeline:**

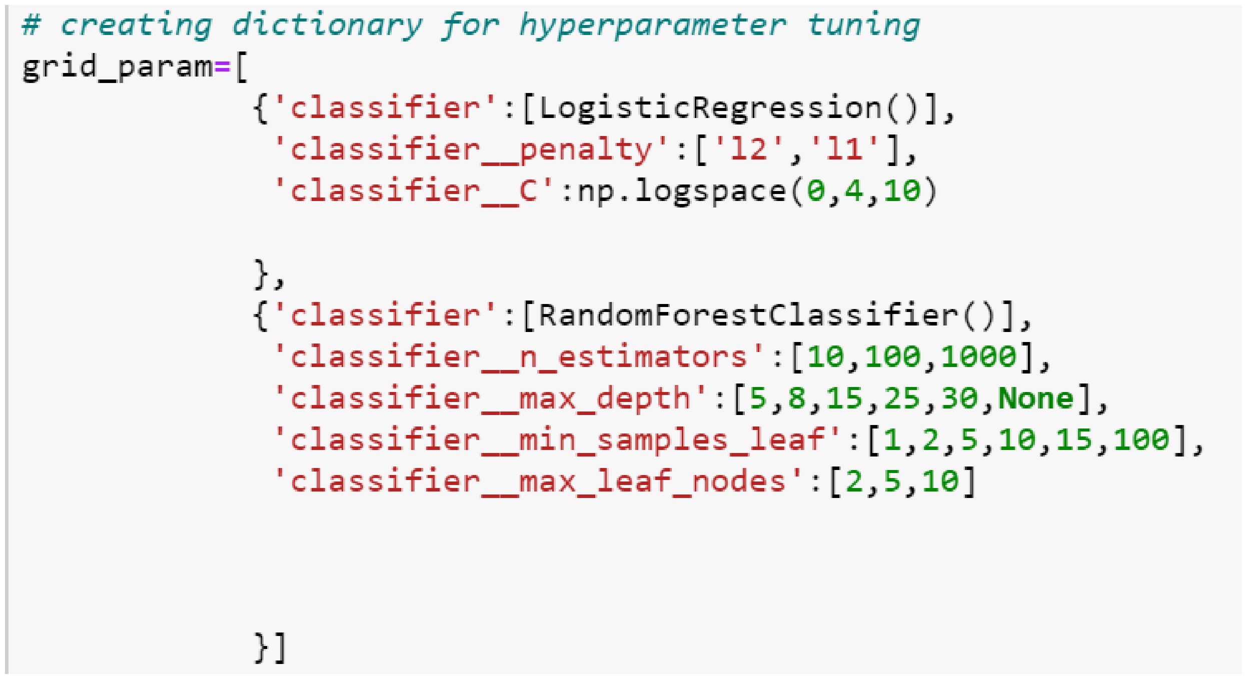
For the implementation of the hyperparameter, I have created a pipeline to combine the model and parameters. The pipeline is nothing, just a process of trying together different pieces of the machine learning process for better results. Pieces mean model and parameter of the model.

**GridSearchCV:**

GridSearchCV is a library function that is a member of the sklearn's model-selection package. It helps to loop through predefine hyperparameters and fit our model on our training set.



```
#create a pipeline
pipe=Pipeline([('classifier',RandomForestClassifier())])
```

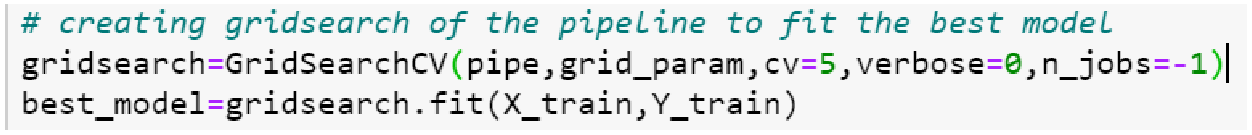


```
# creating dictionary for hyperparameter tuning
grid_param=[
            {'classifier':[LogisticRegression()],
             'classifier__penalty':['l2','l1'],
             'classifier__C':np.logspace(0,4,10)

            },
            {'classifier':[RandomForestClassifier()],
             'classifier__n_estimators':[10,100,1000],
             'classifier__max_depth':[5,8,15,25,30,None],
             'classifier__min_samples_leaf':[1,2,5,10,15,100],
             'classifier__max_leaf_nodes':[2,5,10]



            }]
```



```
# creating gridsearch of the pipeline to fit the best model
gridsearch=GridSearchCV(pipe,grid_param,cv=5,verbose=0,n_jobs=-1)
best_model=gridsearch.fit(X_train,Y_train)
```

Fig10: hyperparameter tuning

The performance of the model depends on different types of parameters, so to get the best result, I have created a dictionary of the hyperparameter for both models. For logistic regression, I have included L2 and L1 as the penalty parameter. The penalty is the comparison of the percentage of zero coefficients of solutions when L1 and L2 are used for different values of C. C is inverse of regularization strength and used three different values for c such as 0, 4, and 10. For random forest modeling, I have used four different parameters such as n_estimators (it has 3 values such as 3, 10, 100, 1000), max_depth (used 5, 8,15,25,30 and none), min_samples_leaf (used 1, 2, 5, 10, 15, and 100) and max_leaf_nodes (used 2, 5 and 10). To iterate through each pair, I have used 5-fold cross-validation.

**Result and Discussion:**

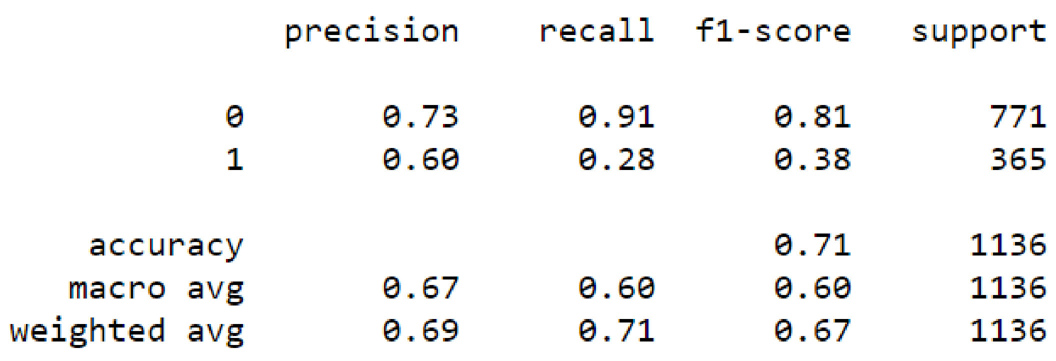
Finally, I implemented both models using hyperparameter and k-fold cross-validation.

And found the following results.

Average precision score: 0.69

Log-loss: 0.34 (I think it is a little higher. Low value of log-loss means better result)

Best model score: 0.79



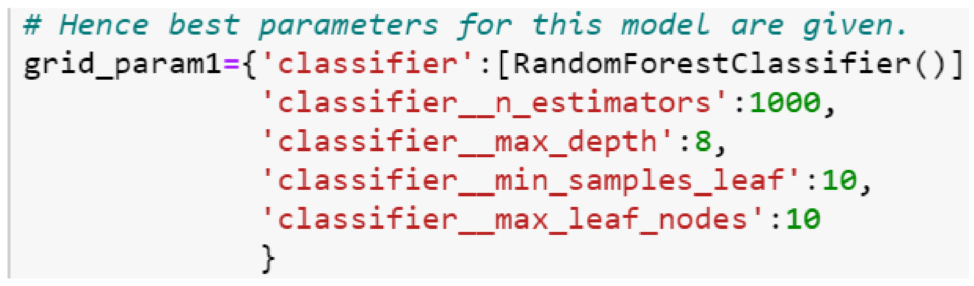
```
              precision    recall  f1-score   support

           0       0.73      0.91      0.81       771
           1       0.60      0.28      0.38       365

    accuracy                           0.71      1136
   macro avg       0.67      0.60      0.60      1136
weighted avg       0.69      0.71      0.67      1136
```

Fig11: classification score of best estimator

Hence, according to the result, the random forest got the best scores and accuracy when n_estimator was 1000, max_depth was 8, min_sample_leaf was ten, and max_leaf_node was 10.



```python
# Hence best parameters for this model are given.
grid_param1={'classifier':[RandomForestClassifier()]
             'classifier__n_estimators':1000,
             'classifier__max_depth':8,
             'classifier__min_samples_leaf':10,
             'classifier__max_leaf_nodes':10
             }
```

Fig12: best parameter and classifier

**Problem/Issues:**

In this project, I have solved the classification problem using two machine learning algorithms implementing various methodologies. While working on this project, I have encountered many problems such as finding the best fit model, handling binary classification problems and tuning hyperparameter, and a little high log-loss, but solved with the supervision of a professor and other classmates.

**Discussion and conclusion:**

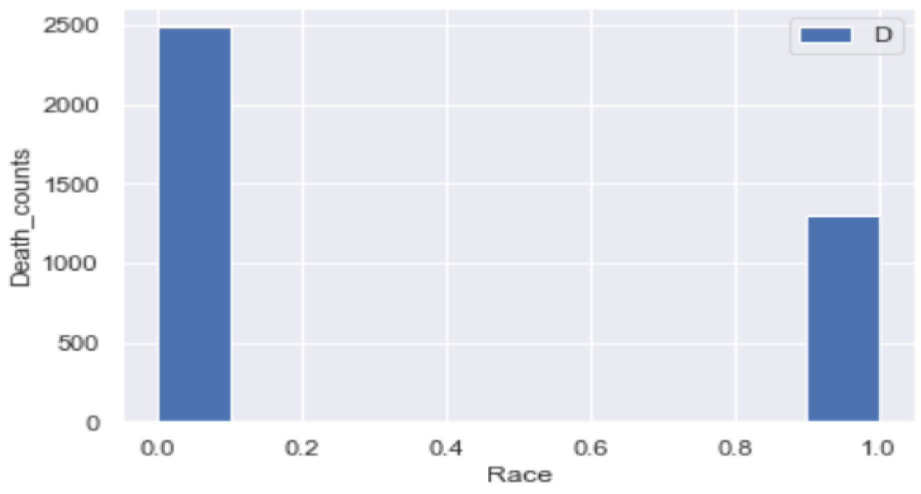
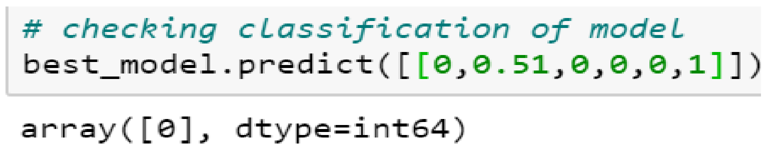


Fig13: death counts based on race

As a result, you can see a histogram of death by race where police killed almost double white people than black. According to my research, I have not found any bias ness between the white and black race. Still, it is a very sensitive case, so we need to deep dive into it to find the actual result because there are so many factors that are being responsible that might affect it, such as population, poverty, income, arrest type arrest count, etc.

Besides that, I have classified the victim based on his/her age, gender, mental illness, threat level, and armed features implementing these two machine learning algorithms. For example



```
# checking classification of model
best_model.predict([[0,0.51,0,0,0,1]])

array([0], dtype=int64)
```

Fig14: testing the model

In the above figure, we can see that the race of the victim is White.

Hence, I have successfully completed this project by classifying the death of the person based on race. I have got 79% model accuracy which is a little less than expected. However, we can improve this accuracy by implementing cross-validation and training more data.

## Acknowledgment:

I can't express enough thanks to my professor Fadi Alsaleem for his continuous support and encouragement. The completion of this project could not have been accomplished without the help of my classmates and beautiful daughter, Prisumsa.

Thanks to my wife Suchan as well. The countless times you kept the children during our hectic schedule will not be forgotten.

## Appendix:

Used libraries

1) Pandas (Open-source data analysis and manipulation tool)
2) Numpy (Main library for computation and array construction)

3) Matplotlib (data visualization and presentation)

4) Seaborn (data visualization and presentation)

5) Sklearn (Machine learning implementation)

**References:**

1) The Guardian Team. (2015, June 2). People Killed BythePolice.https://www.theguardian.com/us-news/ng-interactive/2015/jun/01/the-counted-map-us-police-killings
    a) It provides a brief description of the dataset.
2) Amar Kumar. (2020, June 2). Police Killing. https://www.kaggle.com/amark720/police-killings
    a) I have downloaded this dataset through the above link and understood the features for further analysis.
3) Data Flair Team. (2019, October 10). 11 Top Machine Learning Algorithms used by Data Scientists. https://data-flair.training/blogs/machine-learning-algorithms/
    a) It provides brief descriptions of 11 machine learning algorithms which helps us to choose the best machine learning algorithms for this project.
4) Akira Kanai. (2020, June). Geo mpping: folium. https://www.kaggle.com/recuraki/geo-mapping-folium
    a) Mapping is another important visual method to understand the data. So, this technique helps us to visualize the crime data in map format.
5) Scikit Learn Team. (2020). Machine Learning in Python. https://scikit-learn.org/stable/

    a) Scikitlearn is one of the most important libraries that helps us in data preparation to modeling.
6) Jason Brownlee. (2020, August 19). 4 Types of Classification Tasks in Machine Learning https://machinelearningmastery.com/types-of-classification-in-machine-learning/
    a) This article helps us understand the classification problem and describes the best 4 types of machine learning algorithm methodologies.
7) Evan Lutins. (2017, September 5). Grid Searching in Machine Learning: Quick Explanation and Python Implementation. https://elutins.medium.com/grid-searching-in-machine-learning-quick-explanation-and-python-implementation-550552200596
    a) It clearly explains the implementation of the gridSearchCv in the field of data science. Furthermore, model performance is the ultimate goal of any data science project so, to find the best model among models, we have to use hyperparameter tuning and pipeline. These all concepts clearly explain in this article.
8) Jason Brownlee. (2020, August 15). Logistic Regression for Machine Learning. https://machinelearningmastery.com/logistic-regression-for-machine-learning/

    a) This article describes the logistic regression machine learning algorithms to solve the classification problem.

9) Tony Yiu. (2019, June 12). Understanding Random Forest https://towardsdatascience.com/understanding-random-forest-58381e0602d2
    a) This article clearly explains the random forest classification algorithms and their workflow. Actually, I have learned the concepts and implemented them to classify the police killing data.

10) Bijayanta Roy. (2020, April 5). All about Feature Scaling. https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35
    a) Before implementing the model, we should complete the feature scaling and feature selection process. This article briefly explains this process.