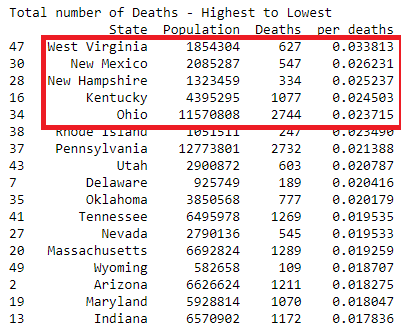
**Data Analysis for opioid overdose fatality summary**:

Model reference:

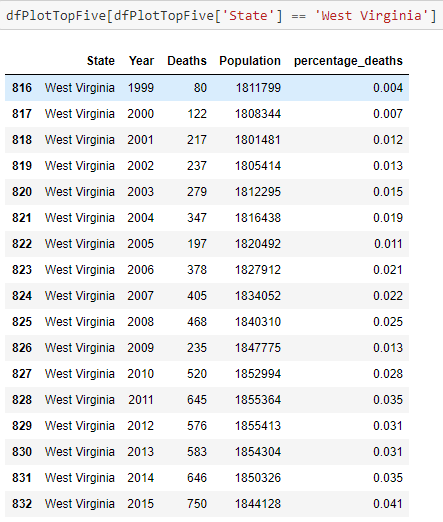
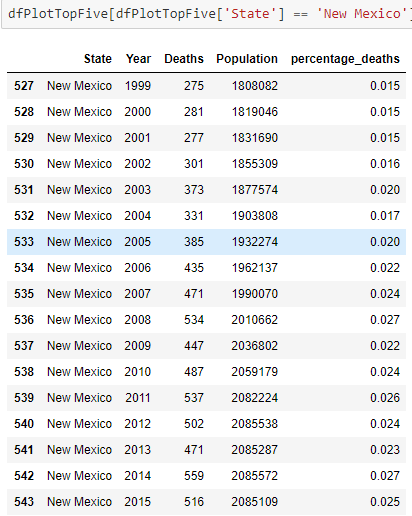
<https://nbviewer.jupyter.org/github/DataScienceKam/health_care_mdoel/blob/master/Opioid_Overdose.ipynb>

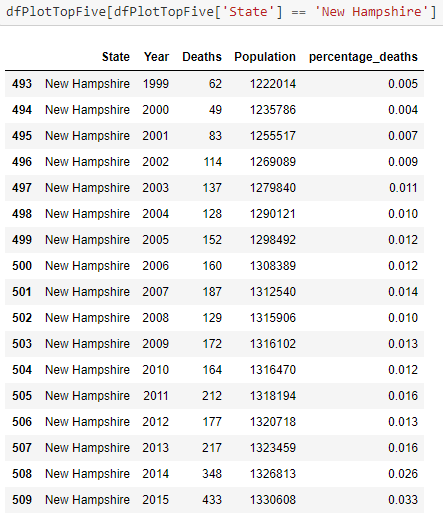
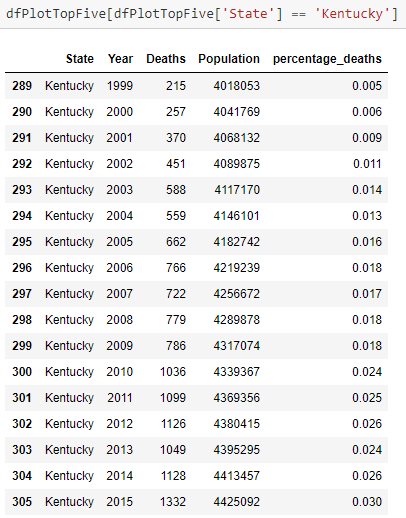
The following observations were recorded by the model datasets:

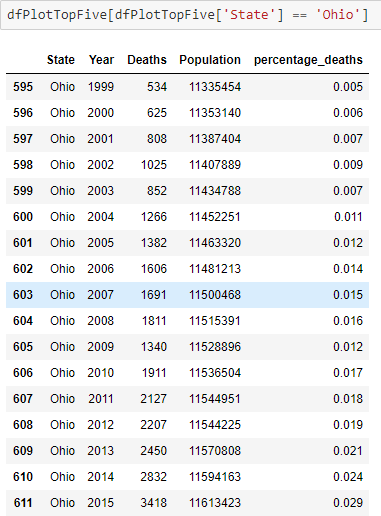
Top 5 opioid overdose fatalities for the year 2014



Investigating further into the top 5 states revealed that there was a steady increase in fatal drug abuse from 1999 – 2015



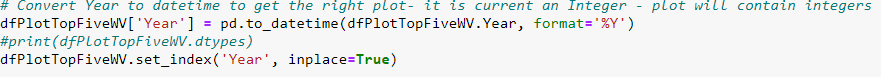
**Time series analysis on overdose fatalities data and projecting the percentage death to a future date using ARMA (ARIMA AND AR did not perform well. There were other issues related to small data size - SVC did not converge in almost all of the models due to a small train – test split. I had to decrease the)**

Train- test split

The following steps to followed to create the test-train split for time series forecasting

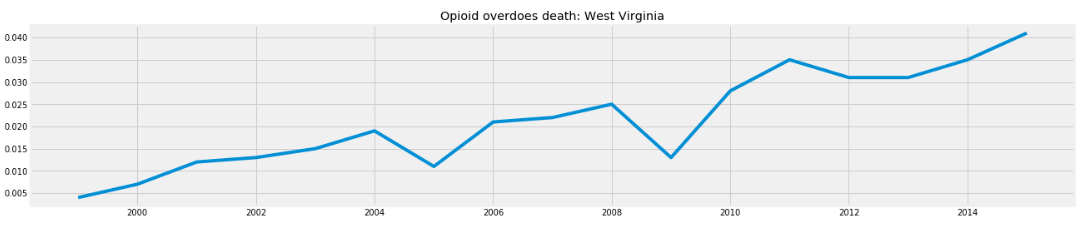
* The Year data type had to be converted from Int to Date
* The dataset was very small, and it was difficult to decide the test-train split. I did however split the test and train data by 70:20

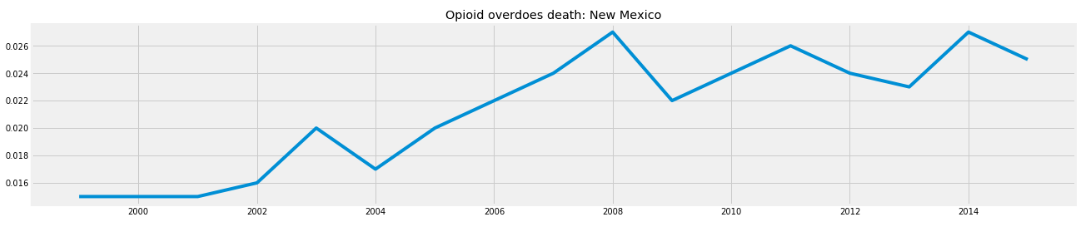
The code for the steps listed above is shown below:

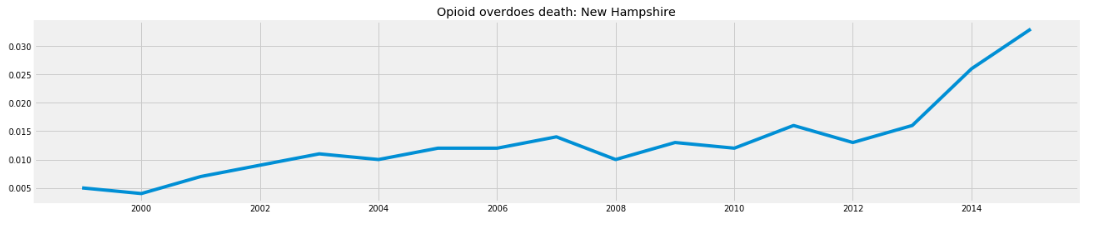


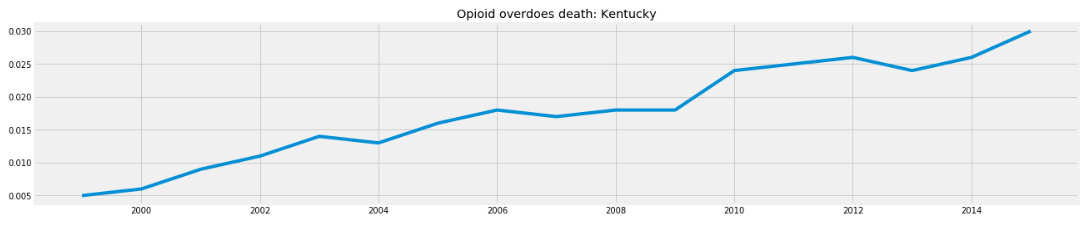


The graph indicated that the fatalities did not follow a gradual increase- there were dips in the graph before a sharp increase in the fatality rates as shown below. This caused the future predictions to be significantly less than the actual value. The ARMA predictive capabilities can be dramatically improved by increasing the size of dataset and converting it to a more granular format by adding month and day values.









Predictions and projection of time line series did not yield accurate results

The red line indicating the predicted score is far less compared to the actual values (indicated by the purple line). Future projections seem to slightly decrease or maintain a steady growth. ***Although we cannot predict/confirm the fact that fatality rates will exponential increase in the near future from the dataset provided (as indicated by the graphs above which seems to dramatically increase from 1999 – 2015), it is obvious that the fatality rate is not going to decrease if the same trend in prescription behavior continues***- ***provided there are no external factors contributing to the decrease In opioid prescriptions such as federal mandates/sanctions imposed on opioid prescriptions***

Accuracy Score:

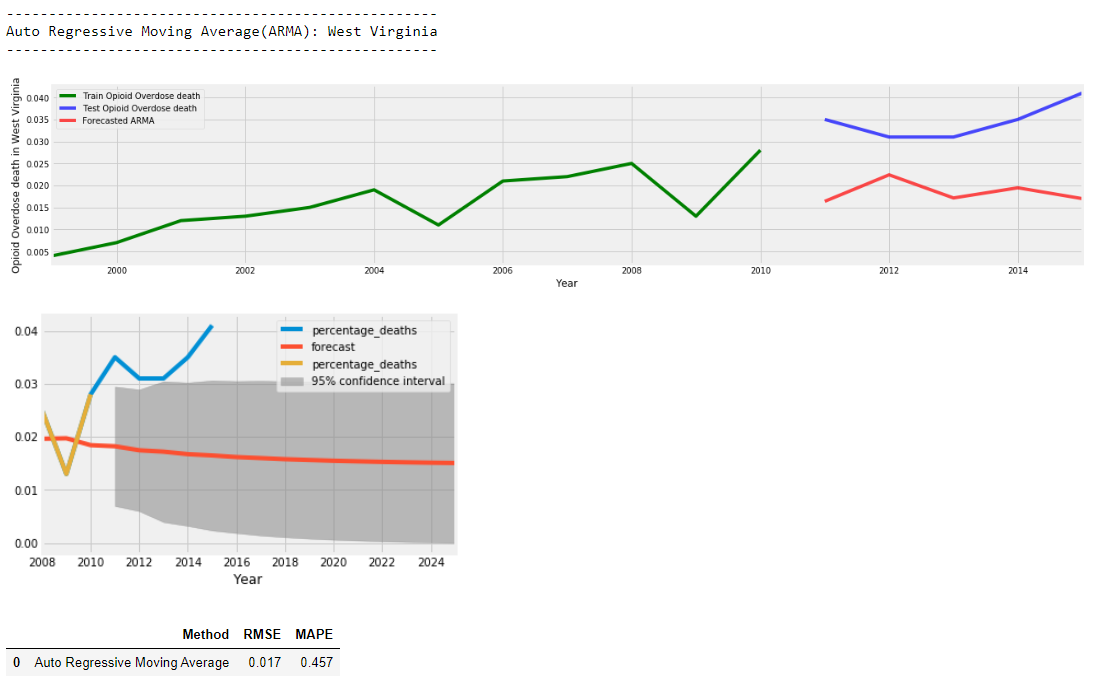
* Root Mean Squared Error: The value indicates the performance of the model using the predicted values against the actual values. The smaller the error the better the model. The best performance of ARMA time series prediction was for Kentucky with the lowest RMSE error indicated by the red line being very close to the purple line. New Mexico also produced a lower RMSE error.
* Mean Absolute Percentage Error: The value indicates forecasting accuracy and was intended to be used in conjunction with RMSE. Like RMSE, as it quantifies the error produced in the model higher values indicate poor performance.

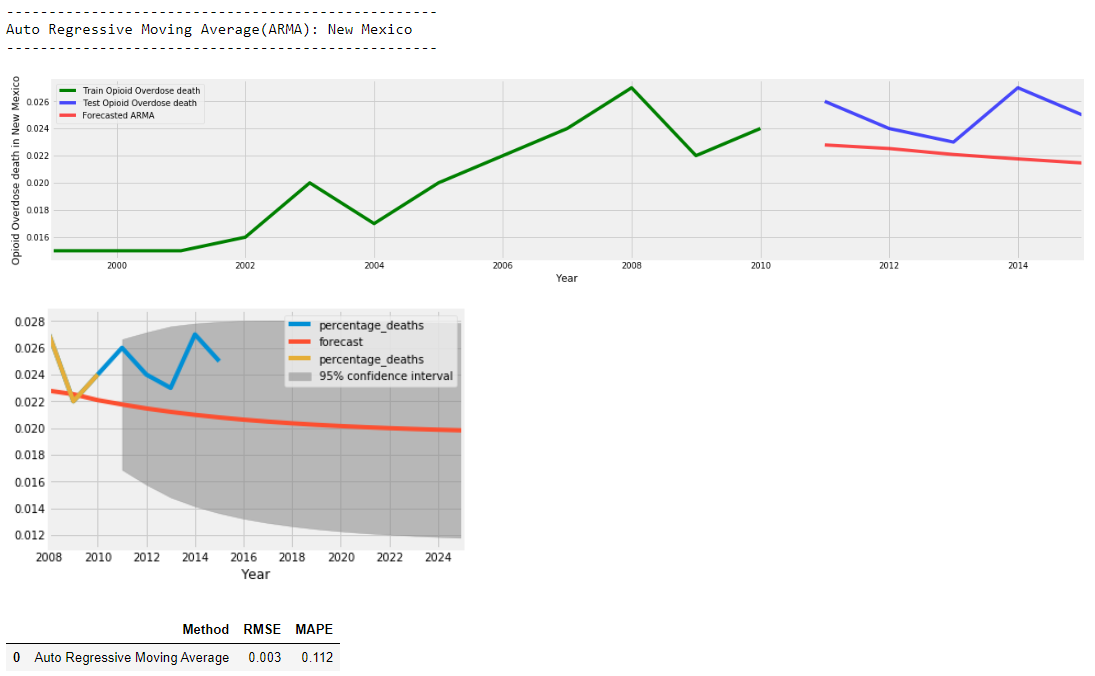
More data with granularity is required for better predictive models in each instance.

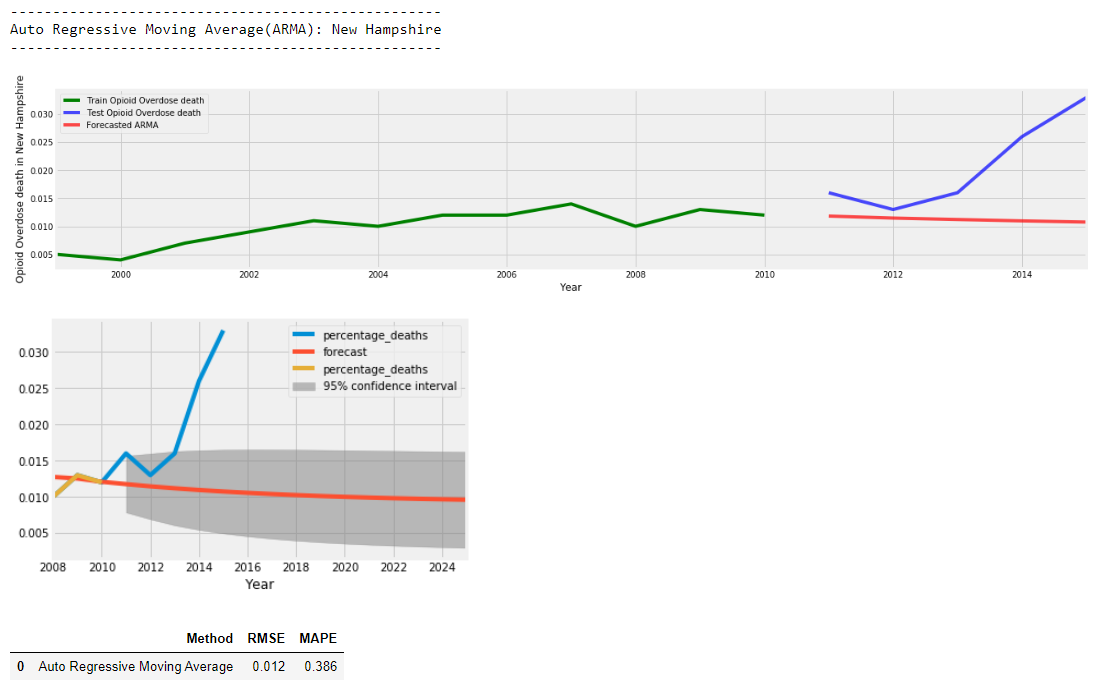
Conclusions that can be derived from the model predictions:

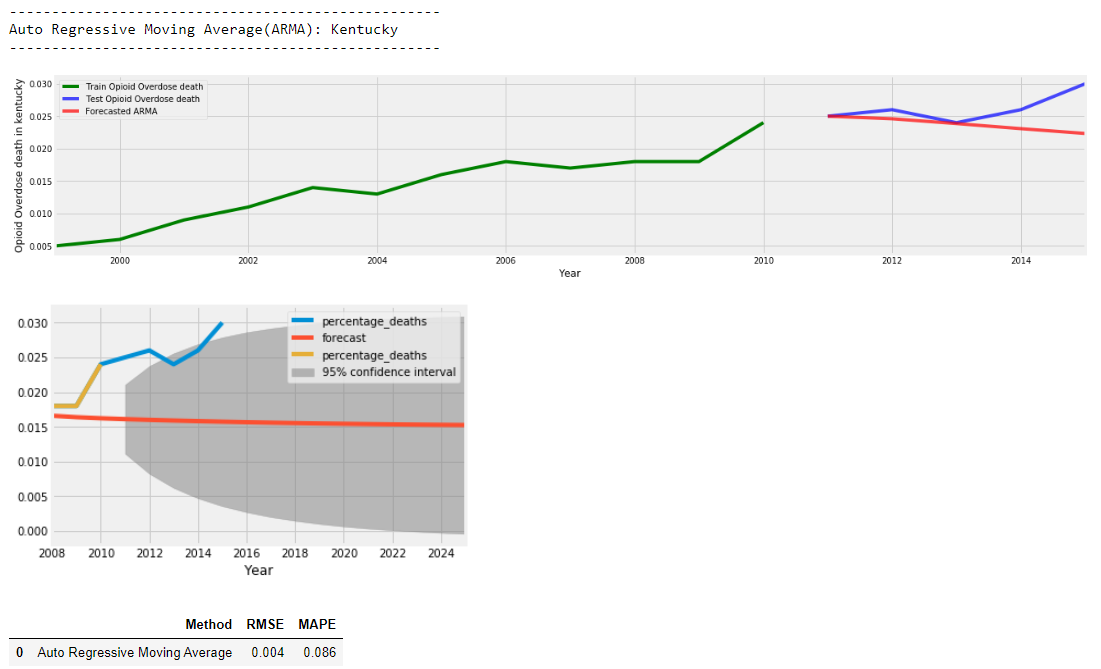
The data is too small to use RMSE and MAPE scores to interpret the final outcome and the selection of an appropriate model. ARIMA and SARIMA were not applicable to the data with the SVC did not converge- data is too small to run SARIMA. ARMA performed much better than AR.

Although the model was not able to accurately predict the growth of fatalities over the years, future prediction does indicate that fatality rate is not going to decrease to acceptable levels without third party intervention.







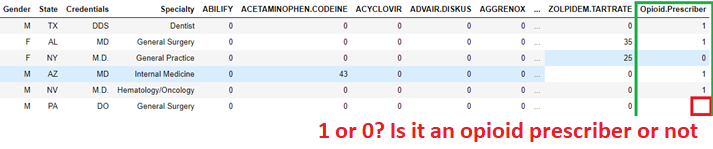


**Opioid overdoes prescription prediction (predicting if a new prescriber is an opioid prescriber or not)**

Identifying the dependent variable/target column

The primary purpose of building the model is to identify the specific prescriber as a possible overdosing candidate given independent variables comprising of columns such as non-opioid drug prescription values, gender, state, and specialty.

In the dataset this column is identified as Opioid.Prescriber



Tuning Process

Selected columns which I think will have an impact on the target column

* Gender
* Specialty
* State
* Non-Opioid prescription drugs (I would assume that these values are relevant to the type of specialty and there is a connection between these drugs and Opioid prescriptions written for the patient. A domain expert such as a pharmacist, nurse practitioner, physician assistant, or a doctor is required for further clarify this point).
* Non opioid drugs include:
  + ACYCLOVIR: Used for herpes virus infections or shingles and used by HIV patients for the treatment of infections
  + DIAZEPAM: Subdues anxiety, muscle spasms, and seizures
  + LAMOTRIGINE: Bipolar disorder
  + IBUPROFEN: Used to quell severe pain
  + FAMOTIDINE: Used for the treatment of ulcers, gastroesophageal reflux disease
  + LIDOCAINE: Relieves pain and numbs skin (analgesic)

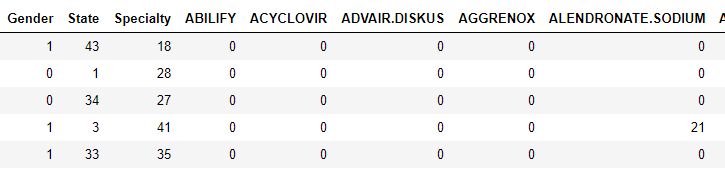
In essence the lists comprises of drugs typically used for the treatment of chronic or acute conditions associated with pain symptoms such as body aches, heart burn, arthritis or HIV to name a few

Rejected columns in the dataset for classification models

* NPI (seems to be noise in the dataset and I don’t think this column will have an impact on the model- there is no useful behavioral pattern that the classifier can pick up to make accurate predictions- this seems to be the unique number of the physician/ the key column)
* Credentials (adds noise to the dataset as it is related to Specialty and seems to be an abbreviated form of the same)
* Opioid drug columns- after running the model I noticed a 99% to 100% accuracy rate on all the classification models. This made me wonder if the opioids columns were causing data to leak in the model accidentally share information between the test and training data-sets. This is my inference the accuracy of which is yet to be determined by experts. I got a more reasonable score in the model after removing the opioid prescription columns from the final dataset

Factorization

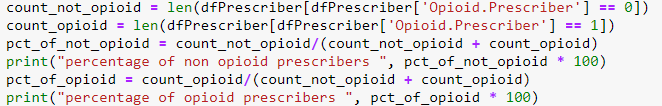
Converting non numeric values to numeric values. Strings were factorized to produce continuous numeric values as shown below:

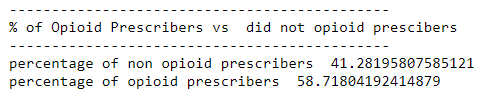


A 30%: 70% test train split was created

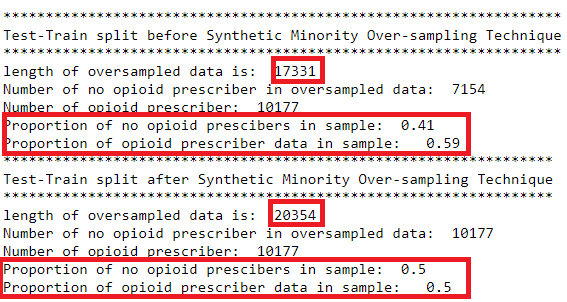


Furthermore, since this is a very sensitive data model, the metrics between opioid prescribers and non-prescribers were first tabulated to see if both the conditions were well represented. 41.28% of the data points consisted of non-opioid prescribers while the remaining values represented opioid prescribers. Machine learning algorithms have trouble learning when one class dominates the other. In this model the machine learning algorithms has more opioid prescribers to learn from vs non opioid prescribers. The score can decrease dramatically with a 30:70 split.





SMOTE technique was used to balance the dataset and ensure that both the options are represented uniformly in the sample dataset



Classification algorithms

Metrics used to gauge the efficiency of the classification algorithms

* **Confusion matrix/ contingency table:**

In the model the following scenarios are useful in classifying the efficiency of the model

* + True Positive: The model correctly identifies the prescriber as an opioid prescriber
  + True Negative: The model correctly identifies the prescriber as not an opioid prescriber
  + False Positive: The model wrongly identifies a non-opioid drug prescriber as an opioid prescriber
  + False Negative: The model wrongly identifies an opioid prescriber as a non-opioid prescriber

***Presumably, I would argue that in our model, it is okay to obtain false positives as human intervention can confirm that the value was not accurate, and the model falsely identified a non-opioid prescriber as an opioid prescriber. It is the false negatives that needs to be lower than the false positive as we do not want to miss any opioid prescribers and misclassify them as non-opioid prescribers. This is similar to airport checks where a passenger is sometimes wrongly identified as a threat. In such circumstances it is okay for the airport authority to intervene and investigate the matter further. However, the reverse can have dire consequences wherein a terrorist is actually missed.***

* **Accuracy Score**: This is the primary starting point of the model which determines the number of correct predictions from all predictions made (# of correct predictions => TP + TN/Total predictions) \* 100. A high accuracy score indicates that the model performed well and can be used for new situations thus a well-defined model will produce a high accuracy score, however this alone does not confirm the validity of the model.

**# Accurately identified opioid prescribers + # Accurately identified non opioid prescribers**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Total number of prescribes and non-prescribers**

* **Recall Score**: Percentage of total results correctly classified by the algorithm. A high recall score is an indication that the algorithm performed well. (TP/ TP + FN)

**# Accurately identified opioid prescribers + Accurately identified non opioid prescribers**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**# of wrongly identified Opioid Prescriber as a non-prescriber**

* **Average Precision Score**: Refers to the percentage of the results which are relevant. A high average precision scare is an indication that the algorithm performed well (TP/TP + FP)

**# Accurately identified opioid prescribers + Accurately identified non opioid prescribers**

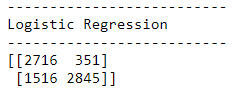
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**# of wrongly identified non opioid Prescriber as opioid prescriber**

* **F1 Score**: This score takes both precision and average into account- a high value is always good

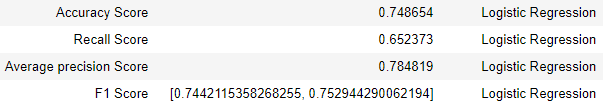
**Linear regression**

Confusion Matrix (a graph of the confusion matrix is available in jupyter notebook)-



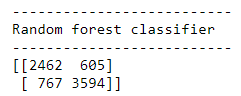
|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP: 2716 | FP: 351 |
| Negative | **FN: 1516** | TN: 2845 |

Evaluation Metrics



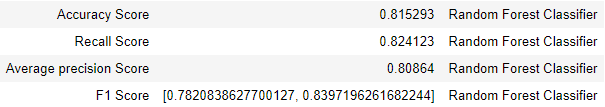
**Random Forest Classifier**

Confusion Matrix (a graph of the confusion matrix is available in jupyter notebook)-



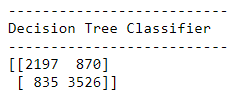
|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP: 2462 | FP: 605 |
| Negative | FN: 767 | TN: 3594 |

Evaluation Metrics



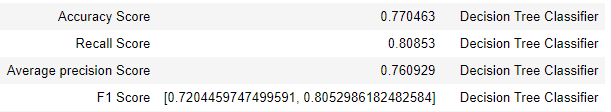
**Decision Tree Classifier**

Confusion Matrix (a graph of the confusion matrix is available in jupyter notebook)-



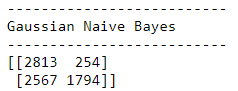
|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP: 2197 | FP: 870 |
| Negative | **FN: 835** | TN: 3526 |

Evaluation Metrics



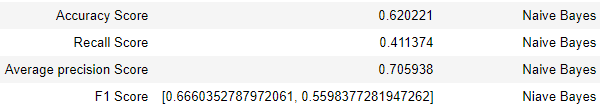
**Naïve Bayes**

Confusion Matrix (a graph of the confusion matrix is available in jupyter notebook)-



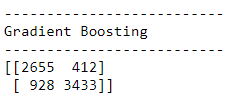
|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP: 2813 | FP: 254 |
| Negative | **FN: 2567** | TN: 1794 |

Evaluation Metrics



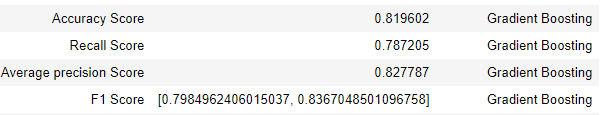
**Gradient Boosting**

Confusion Matrix (a graph of the confusion matrix is available in jupyter notebook)-



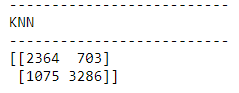
|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP: 2655 | FP: 412 |
| Negative | FN: 928 | TN: 3433 |

Evaluation Metrics



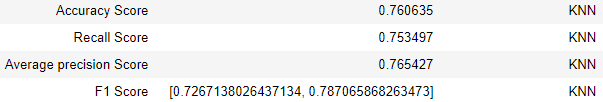
**KNN**

Confusion Matrix (a graph of the confusion matrix is available in jupyter notebook)-



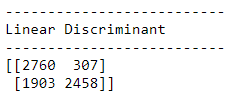
|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP: 2364 | FP: 703 |
| Negative | **FN: 1075** | TN: 3286 |

Evaluation Metrics



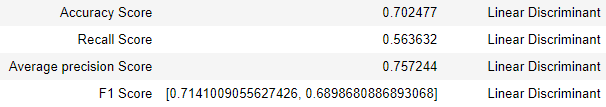
**Linear Discriminant**

Confusion Matrix (a graph of the confusion matrix is available in jupyter notebook)-



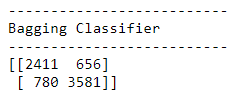
|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP: 2760 | FP: 307 |
| Negative | **FN: 1903** | TN: 2458 |

Evaluation Metrics

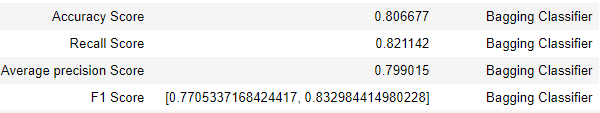


**Bagging Classifier**

Confusion Matrix (a graph of the confusion matrix is available in jupyter notebook)-



|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP: 2411 | FP: 656 |
| Negative | FN: 780 | TN: 3581 |



**Stochastic Gradient Descent Classifier**

Accuracy Score

I did not go beyond this point as I did not get a good accuracy score

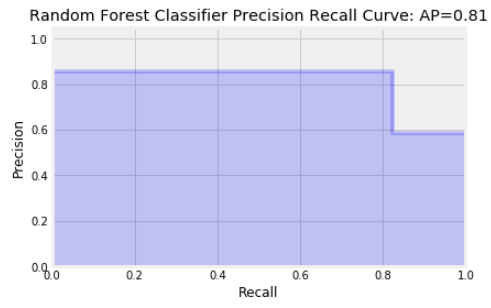


Conclusion

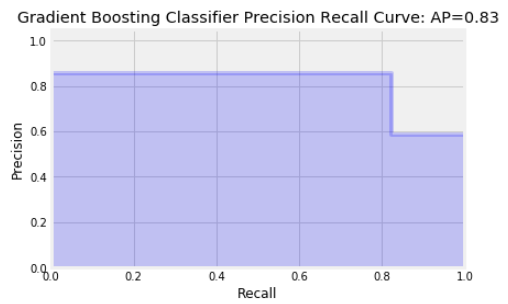
Based on the evaluation metrics obtained, I would recommend the following 3 algorithms -as they have a highest accuracy rate and seemed to perform well in both Recall and Average precision score amongst the other models- to be used for unknown/new data points to predict Opiod.Prescriber binary value, 1 indicating the fact that it is an opioid prescriber while 0 indicating the fact that it is not an opioid prescriber.

The plot below shows the precision-recall curve to help determine the success of the trial (recall => performance measure of the whole positive part of the dataset and precision => performance measure of positive predictions)

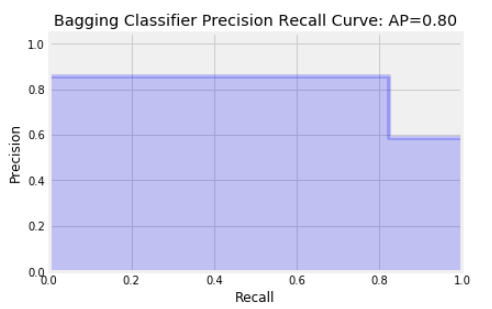
* ***Random Forest***



* ***Gradient Boosting***



* ***Bagging Classifier***



Interpretation of the Precision Recall Plot

* A value below 0.50 is considered poor
* A value above 0.50 is considered a good model
* 0.50 indicates a model with is neither good nor bad but in general as a rule of thumb a 50% average model is not considered a well performing model in my opinion
* The more successful a model is, the less false positives and false negatives and more true positives and true negatives.
* The precision-recall score represents this total, and the precision-recall curve helps to visualize it.
* In the above plot all the 3 models have a high precision-recall score and would therefore be the models that I would suggest for future use