Analyzing New York Police Racial Profiling With A Machine Learning Approach

Course Project - Machine Learning

Àlex Martorell i Locascio
Louis Van Langendonck
Facultat d'Informàtica de Barcelona

Stop, Question and Frisk Locations in Harlem, NYC
BLACK
WHITE
OTHER
40.8150
40.8125
40.8100
40.8075
40.8050
40.8025
40.8000
40.7975
−73.970
−73.965
−73.960
−73.955
−73.950
−73.945
−73.940

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

June 2022

Contents

1 Introduction

1.1 Description of the data and goals of the project

For this study, the 2020 New York City Police Department (NYPD) Stop, Question and Frisk Data Set is used. It can be found at `https://www1.nyc.gov/site/nypd/stats/reports-analysis/stopfrisk.page` [1]. The data set considers a practice in New York City called "Stop, Question and Frisk". This widely known procedure consists of an officer stopping a person whenever there is a suspicion that a crime is being committed, sometimes followed by an arrest or a frisk (searching for hidden items by feeling the person's clothing). According to NYPD regulations, for each stopping action a Police Officer completes, he or she must report it by filling in a UF-250 form [2]. The filled out forms of the year 2020 constitute this data set.

The "Stop, Question and Frisk" data set is very extensive, and contains detailed information for each single stop, question and frisk instance. General things covered are the location of the stop (Street, borough, police sector...), the suspect description (Age, height, weight, race, demeanor...) and some general binary stop specifics (Physical force used, weapon found..). In total, there are 9544 rows and 83 columns (72 categorical, 10 numerical and 1 time stamp variable).

The **goal** of this project is to try and predict race based on the different categorical and numerical variables available. The reason behind this choice is two-fold: Firstly, correlations between variables were assessed (as explained in the Project Proposal) and some already insightful results were obtained for some variables in the data set. Secondly, the debate on Race in America has existed since its foundation as a country, and recently has been back in the news again regarding Five-O [3] brutality claims. There are many studies that have already targeted Police data and found a clear racial bias in some of their actions. This is why it may also be interesting to study further correlations, by looking at how predictive models are able to distinguish the stop and frisk of a Black person, a White person or a person of different origin (f.e. "Black stops" have higher probability than "White stops" at nighttime).

1.2 Important remarks

It is not a surprise to any reader that the American Police system has been under scrutiny for many years. This has also affected the New York Police Department, object of this current study. In 2014, the Stop Question and Frisk policy was ruled to be Unconstitutional based on 4th Ammendment grounds. [4]. This means that such policy has gone through improvements and if the current dataset (Dated from 2020) is compared to some from previous years (i.e. 2010) not only are there less cases, but they happen to be less racially-motivated. This has to be taken into account when comparing results with previous studies undergone before the court ruling in 2014.

1.3 Related work

There is a substantial amount of literature available that has made this particular topic and even this same data set the core of its research. Three relevant articles are selected which were published in an 8 year span. They focus on applying Machine Learning algorithms to the Stop, Question and Frisk data set. In [1] we find a discussion on how policing strategies are experienced differently by genders. A Logistic Regression model is used to conclude the Black males have a higher probability of being stopped and frisked. Moreover, an article just published in June 2022 [2] (at the time of writing) also is focused on finding underlying patterns in Stops and Arrests. It focuses specifically on bias metrics and bias mitigation techniques. Last, [3] focuses on the Stops triggered by CPW (Criminal Possession of Weapon). It studies the *ex-ante* probability that the detained suspect actually has a weapon, which was revealed to be less than 1%. Also, Blacks and Hispanics are found to be dis-proportionally targeted.

[1]Note that additional information valuable for this project can be found in this site.
[2]`https://www.prisonlegalnews.org/news/publications/blank-uf-250-form-stop-question-and-frisk-report-worksheet-nypd-201`
[3]Five-O is US-street slang for "Police"
[4]`https://civilrights.org/edfund/resource/nypds-infamous-stop-and-frisk-policy-found-unconstitutional/`

2 Data exploration

2.1 Pre-processing tasks

2.1.1 Train Test Split

Before any decision or transformation is made on the data set, it is split into a 80 % / 20 % train- and test set. With 9544 rows, the data set is considered large enough to warrant having the test set size at only 20 %.

2.1.2 Initial Inspection and Cleaning

Initially some data exploration is done. It is quickly found that some numerical values are wrongly expressed as categorical values. These variables (suspect reported age, suspect weight and suspect height) are thus converted into numerical values. A similar simple transformations is made to 'stop frisk time' to transform it from a categorical time description to a float. Next all variables are plotted, using histograms for both numerical (with small bins) and categorical values. These can be found in the appendix in Figures 6.1 and 6.2. A few conclusions can be made from data exploration and these plots:

- Most categorical values are rather imbalanced in their distribution and might benefit from collapsing some category levels. The same applies to the target variable, suspect race description. It contains too many subdivisions, of which the occupation is unbalanced (fe. having a category level for both white and white hispanic with the latter not containing nearly as much instances in the data set). Some thoughtful collapsing will be needed here too. Luckily it does not contain missing values.
- The data needs cleaning: naming conventions for categories are not always consistent (fe. sometimes 'Y'/'N', other times 'Y'/'(' or 'Y'/'null'). Moreover, some data mistakes are present (fe. suspect being 2.5 meters, suspect having age 0, etc.).
- Missing values are generally present, again with inconsistent naming conventions.
- Some numerical categories may contain outliers.

A second part of data exploration is gaining familiarity with the features by the means of visualization. As a lot of categorical features are present, cross-tabs can be used to find some interesting relations. For example, the relation between whether or not a suspect is arrested and the NY borough the stop took place can be found in Figure 2.1. Some similar interesting visualizations can be found in the Appendix in Figures 6.3- 6.5.

2.1.3 Missing Values

As already mentioned above, in terms of missing values, some data cleaning is needed because often different syntax is used to indicate a candidate missing value, such as: `(null)`, `XXX`, `(`, `nan` or a blank space. Note that this happens both across numerical and categorical variables and does not always indicates a missing value, making it necessary to approach these case by case.

For categorical variables, each one is looked at separately to see what the candidate missing value means. Two examples are given to illustrate the process. In 'suspect hair color', values like 'ZZZ' are clearly missing values. In contrast, 'physical force verbal instruction flag' (a binary variable) has values 'Y' and '(' where the '(' here clearly signifies 'N' instead of missing. All the missing values are initially set to NA.

The only numerical features containing missing values are 'suspect reported age', 'suspect weight' and 'suspect

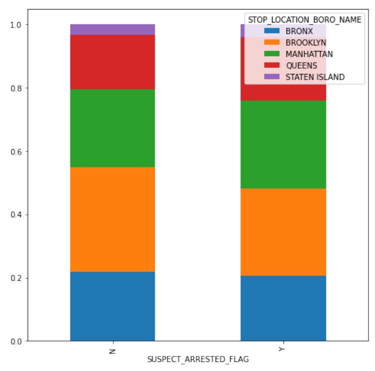


Figure 2.1: Cross-tab showing relation between stop borough and arrested flag.

height'. In these cases, missing values are present in the form of blank spaces or defective data (like impossibly large people or people with the age of 0). These are both set to NA.

Next, rows containing a lot of missing values at once are looked into. It is found that most missing values are distributed across the data set, possibly suggesting a underlying missing-at-random mechanism (MAR). However, a few rows (less than 3 %) have more than 6 (up to 15) missing values at once. These are considered uninformative and removed from the data set.

It is chosen for most categorical variables with a large amount of missing values to assign them to a new level called 'unknown'. This often makes sense semantically, as these missing values are probably the result of the police officer not filling in that variable. Some other variables with a significant amount of missing values (more than 30 % missing and less than 5 % Chi2 correlation to the target variable) are just removed from the data set as they are considered not informative in the first place (fe. 'stop location apartment', a variable indicating the floor of the apartment where the suspect is arrested. It is considered a uninformative detail). After these operations no more categorical variables have missing values.

There are a few remaining missing numerical variables (7.6 % of age variable, 4.0 % of weight variable and 3.1 % of height variable). Variables age, weight and height are found to be significantly correlated with respect each other and some other variables (suspect build type, race, hair color...). This is displayed in table 2.1. Therefore it is chosen to impute the missing values using a `KNNImputer` with `n_neighbors = 3`. This imputation converges and is considered to not introduce too much bias into the data set because not too many values are imputed and highly predictive variables are used.

Race	Height	Weight	Body Build Type	Hair Color	Zip Code	Patrol Borough
0.12	0.20	0.30	0.17	0.37	0.24	0.14

Table 2.1: Correlation of Age with other variables

After these operations, no more missing values are present in the data set.

2.1.4 Outlier Treatment

For the categorical variables, the imbalance is handled by collapsing minority levels, which is further explained in Section 2.1.5. As the data set contains mostly categorical variables, this already concludes most outlier treatment. The remaining numerical variables are: Height, weight, age, time, stop duration minutes and observed duration minutes. For height, weight and age, outliers were looked into earlier during missing values treatment in Section 2.1.3 as these are considered mistakes in the data. These values were set to NA and imputed. For the remaining numerical values, extreme outliers are found for both the 'stop duration minutes' and 'observed duration minutes' as is displayed in Figure 2.2. As there are occasions of stops and frisks that probably indeed took a very long time, such that these values are not considered mistakes and might contain valuable information in prediction. However, the imbalance extreme outliers create might negatively impact some predictions, clustering, etc. Therefore, for the stop duration, instead of deleting these rows, it is chosen to manually rescale all extreme outliers to the top value of the mild outliers, thus retaining the fact that the time of the stop was indeed very long, but decreasing the amount of imbalance created by the extreme outliers in model training and clustering. The result of the rescaling can be found in Figure 2.3. For the observed duration minutes, it is found that all values except outliers are listed as 1, indicating an immediate stop. Therefore it is decided to instead of removing outliers, convert the variable in to a binary categorical variable called 'Immediate stop flag' with levels 'Y' for all immediate stops and 'N' for the rest. The result can be found in Figure 2.3.

2.1.5 New Variable Derivation

One potentially interesting column for prediction might be 'demeanor of person stopped' which is what the police wrote down to describe the behaviour of the suspect. As it contains 7570 unique values, it can be considered natural language data. To be able to use is as a predictor, a simple NLP analysis, using the `ntlk`-library, is applied on the variable: First, the sentences are pre-processed using a tokenizer and lemmatizer. Subsequently the stop words are removed and the remaining descriptions vectorized. Next, KMeans is applied on the training vectors for multiple possible k's. Looking at the inertia plot displayed in the appendix in Figure 6.10, it is clear that the inertia is

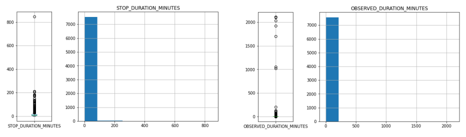


Figure 2.2: Stopped duration minutes (left) and observed duration minutes (right) before outlier treatment

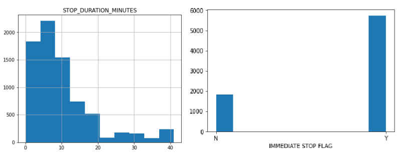


Figure 2.3: Stopped duration minutes (left) and observed duration minutes (right) after outlier treatment.

minimized from 11 clusters on, such that it can be considered a good choice of k. The cluster for corresponding to each row is stored as a new variable 'kmeans demeanor'. Looking at the cluster results, it can be found that the clustering made semantically senseful decisions (fe. all sentences containing the word calm are together in a column, thus resulting in a 'calm' category level). A mapping of the semantic meaning of each cluster can be found in the Appendix in Table 6.1. In the test set, the fitted vectorizer and fitted kmeans model from this training set is used to transform and predict for each test demeanor description to which kmeans group it belongs. It is verified that the labels for the test set are indeed classified according to the ones from the train set.

Next, level collapsing is looked into, in order to improve on categorical imbalances. Two case-dependent approaches are maintained. First, quite a lot of categories display a few well represented columns, followed by a selection of barely occupied ones. It is then chosen to keep the well represented ones and collapsing the rest into a 'other' columns. An example to illustrate this is: the 'Issuing officer rank' has 6379 instances of POM (police officer male), 801 of POF (police officer female), 138 of SGT (sergeant) after which the 10 remaining levels each contain only a handful of cases (from 1 up to 66). It is chosen to collapse these into the 'other' column. A similar transformation is applied on 'issuing officer rank'.

Secondly, some collapses are done based on semantic similarity. For example for the target variable 'suspect race description'. Collapsing is mandatory because of the present imbalance will make for difficult prediction (4292 Black, 1648 White Hispanic, 675 White, 646 Black Hispanic, 185 Asian / Pacific Islander, 109 Middle East, 15 American Indian). However just throwing every minority class into a 'Other' class doesn't make sense. Instead, 'Black' and 'Black Hispanic' can make up the 'Black' class, similarly for collapsing 'White' and 'White Hispanic' into 'White'. The remaining classes are collapsed into 'Other'. This is warranted as the goal of the project is mainly looking at the difference of treatment of the most present race groups in New York (Black and White as a majority, all other race groups as a third group). Similar transformations are applied on 'suspect body build type', 'suspect eye color' and 'suspect hair color'.

2.1.6 Pipelining

Finally, the whole pre-processing pipeline should be applied on the test set, splitted at the very beginning of the process. In order to do this, a function `'pre-process'` is built applying all previous steps in a similar manner and order. The final obtained train and test set are both written to `.csv`-format, ready to be used for model training (train set) and final model evaluation (test set).

2.2 Feature selection and extraction

The goal is to predict race, which is a categorical variable. Recall that it has been collapsed into three levels: "Black", "White", "Other". It can be seen at plain sight that not all variables contribute to the prediction. Plus, using too many variables can lead to overfitting of a model, reduce its accuracy, and result in higher training time. Hence, the correlation between the target and the predictors has to be assessed such that a informed decision on which variables to keep can be made.

Observe that two kinds of profiling for the race variable are ought to be conducted: First with its categorical predictors and later its numerical ones. Identical methods should not be used since specific statistical tests are available for each case.

2.2.1 Statistical Tests

For categorical predictors, the global relationship between race and each predictor can be checked using the homogeneity chi-squared test [4]. Information of value can also be obtained by checking the relationship between race and the levels of each predictor (i.e. If the different boroughs in New York City affect the race variable) using the z-test for homogeneity of proportions [4]. Nonetheless, it also possible to apply the chi-squared test again. Observe that the threshold for the p-value has to change, because multiple hypothesis are being tested at the same time. The Type I error at α is capped to α/m, where m is the number of hypothesis being tested. This is called the *Bonferroni Correction* [5].

Observe that some of the variables with high correlations are nonsensical. 'Street Name' and 'Address' are listed as high correlation with the dependent variable because in many cases they are unique instances.

The Kendall coefficient can be used in a non-parametric test to determine the correlation between a categorical response variable and numerical predictor variables. It measures concordance between them [6].

This first phase of Feature Selection reduces the amount of categorical variables (79) after Preprocessing to 27. For the models that will be used, except for the Multilayer Perceptron, this might still be far too many variables. Further techniques are assessed to reduce the number of variables.

2.2.2 Extra Trees

The `ExtraTreesClassifier` from `sklearn` (which will be used in Modelling , c.f. Section 3.3) can be used to retrieve feature importance for the remaining categorical variables. This subset of 27 categorical variables has to be One hot encoded in order for the Classification to run. Hence, a sparse matrix with 137 columns is obtained. Integer Encoding does not make sense here because in some variables an order would be generated and there is not any. The following conclusions can be extracted from the scores obtained: Non flag variables such as 'Hair Color', 'Body build', 'Eye Color', 'Sex', 'K-means demeanor' get high feature importance scores. The Offense committed and the location (by borough). Regarding flag variables, Background Circumstances, Searched, Frisked are kept, among a few others. Find in Figure 6.6 the Best 10 features according to this classifier.

The final choice, which will serve for all clustering and modelling techniques is to keep 13 categorical variables (plus the desired prediction) and 5 numerical variables. In total, 19 columns.

2.3 Clustering

The objective of this part is to gain deeper insight into the data set by K-Means Clustering. For this to be applied, an strictly numeric variable data set is needed. The data set for this project is mostly categorical, so MCA (Multiple Correspondence Analysis) is tested (Using `prince` package) not only to reduce dimensions but also to have numerical variables explaining the categorical features, which can come in handy for some algorithms. However, categorical variables will be prioritized for modelling if the algorithm accepts categorical variables: The overall accuracy is likely to be higher as MCA generally only captures a subset of all variance.

The first 10 dimensions explain approximately 25 % of the whole inertia. Observe that the first dimension is referred to as 0, and the last one as number 9. In Figure 2.4 it is clear that large values for Dimension 0 correspond to arrested suspects charged with a specific crime, since in the negative values for this dimension the label 'Suspect Not Arrested' is present. Dimension 0 also separates the 'Searched' flag (Positive values for Yes and Negative for No). Dimension 1 clearly separates the Flag variable 'Background Circumstances Violent to Crime' (i.e. Was violence visible before the PO conducted the stop), as well as the 'Frisked Flag'. Location wise (borough where stop is conducted), Dimension 1 also has positive values for Manhattan and negative for The Bronx.

Observe also that in Dimensions 0 and 1 almost all 'K-Means Demeanor' labels are close to the center. This is an exception for instances which are equal to 2 (described as nervous) and 5 (described as agitated), located to the left and right of Dimension 0 axis and have negative values for Dimension 1. Dimensions 2 and 3 separate based on the type of 'Suspect Arrest Offense', as shown in Figure 6.8 in the Appendix.

The function that computes the K-means clustering is run for several values of $K = 2, 3, 4, 5, 10, 20$. The Calinksi-Harabasz (CH) index is a ratio of the separation between the cluster in comparison to

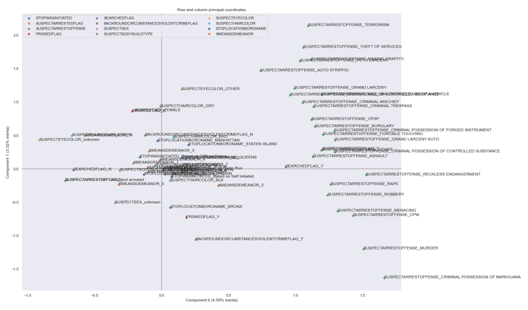


Figure 2.4: MCA Dim 0 and 1

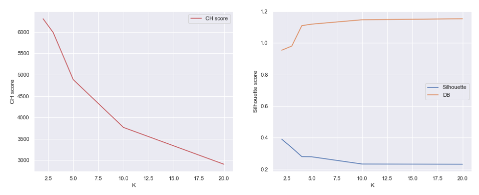


Figure 2.5: Clustering metrics for different values of K

the separation within the cluster. The Silhouette Coefficient (S) runs a mean on the results obtained subtracting the Mean Intra Cluster Distance and the Mean Nearest Cluster Distance. Finally, the Davies-Bouldin score measures the average similiarity of each cluster with its most similar cluster.

Figure 2.5 shows the values of the three metrics for the tried K. Note that a high CH value means bigger separation between clusters, which is reached in $K = 2$ for the CH metric. The best value for the Silouhette and DB coefficients are 1 and 0 respectively. Therefore, all indexes seem to converge to the $K = 2$ as the best number of clusters.

	Rows	Stop Min.	Age	Height	Weight	Stop Hour	Dim 0	Dim 1
Cluster 0	4806	10.255	27.4	5.597	149.8	13	0.006	0.007
Cluster 1	2764	9.975	33.9	5.727	199.6	12.5	-0.010	-0.013

Table 2.2: Some of Global Means for Both clusters

To understand the 2 different clusters, the global mean of the variables is calculated. Such results can be seen in Table 2.2. Cluster 0 contains 4806 (63.49% of the data) instances where as Cluster 1 has 2764. Observe some significant differences between these means. Individuals in cluster 1 are taller, heavier and older.

The Gaussian Mixture for models with Expectation Maximization is a reasonable thing to try when we have data that is not gaussian (i.e. does not come from a normal distribution), which is clearly our case. However, similar results are obtained. $K = 2$ gives the highest CH score, the Silhouette index is highest, and the DB closest to 0.

3 Models

3.1 Introduction

The goal is to predict the 'suspect race description' variable. In the pre-processing steps, the collapsing of the race variable is assessed. This is done to help the training of an ML model. If some categories have too few occurrences, the models could easily overfit. The final distribution is laid out in Table 3.1.

Black	White	Other
4938	2323	309

Table 3.1: Distribution of the Race Variable

One must first state the nature of the data set in order to be able to comprehend the precision metrics of any ML algorithm. Note that the data set is strongly imbalanced, as it can be seen in the distribution of the variable to predict. [5]

[5]`https://www.census.gov/quickfacts/newyorkcitynewyork` reveals an imbalance between the race of the stopped, questioned and frisked and distribution of population by race in New York City.

3.2 Resampling and Validation Strategy

As already noted, a 80 % - 20 % train-test split has been done at the very beginning of the pipeline, even before data exploration and pre-processing. It will eventually be used to test the generalization of the best optimized model, trained on the training set. However, to validate which optimized model performs best, some train-validation splitting has to applied on the training set too. Even more, to optimize a model, cross-validation has to be considered in order to decrease overfitting and data leakage when hyper-optimizing parameters.

When choosing a splitting- and validation protocol, the imbalance of the target data has to be taken into account (see Table 3.1). To ensure the train- and validation set contains a similar distribution, a stratified 80 % - 20 % `train_test_split` is applied.

Now, the cross-validation strategy has to decided upon. For the same reason as mentioned above, a stratified K-fold cross-validation is used. Because of the sufficient size of the data set, 10-folds are used when validating. To further reduce bias in the model, the validation is repeated three times and the data shuffled for each split. The final result is a split given by
`RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=14, shuffle = True)`.

Next, the evaluation metric appropriate for model selection in imbalanced multi-class classification is looked into. The average F1-score is a viable option, as it takes into account both the precision and recall. In this case, two options are considered: macro F1, which calculates the plain average F1-score without weighting according to class size, and micro F1, which calculates a proportion of correctly classified observations out of all observations. The former implies optimizing a model in order to predict all three classes equally (less bias, lower overall accuracy) while the latter will opt for models that are good in predicting in general, often leading to preferring the majority of cases (higher bias, higher overall accuracy). In the case of this project, it is chosen to work with the macro-F1 score because it might result in a more general solution and insight instead of only the majority class(es) ('Black' and maybe 'White'). However, to gain insight on per-class performance, confusion matrices are used.

Finally, note that all splits, validation protocols and models make use of seeds in order to improve reproducibility. The arbitrary number 14 is chosen for this.

3.3 Decision Trees and Random Forests

In this section, the different methods in the Decision Trees and Random Forests are covered. Finally, Extra Tree Classifiers are tested.

A visual output of the first three levels of depth in the Base Decision Tree can be seen in using the graphviz package in Figure 6.9 located in the Appendix. The initial splits have found Hair Color, Eye Color, Height and Age to be the more important features, as was predicted by Feature Selection Techniques (cf. Section 2.2)

The base/default model returns an accuracy of 0.616, a Macro F1 score of 0.439 and a Weighted F1 score of 0.622. The weighted F1 score calculates the F1 score per label and it averages it by the weight of each category. Accuracy is not much of a reliable metric, since it is a global measurement that does not take classes into consideration and returns a high percentage. This is why balanced accuracy is a more useful metric because it involves avoids inflated performance estimates on imbalanced datasets [7]:

$$\text{balanced-accuracy } = \frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$$

In the default model the Balanced Accuracy is 0.444.

First observe that such tree is clearly overfitting, because it predicts the training data perfectly, with an F1-score equal to 1. This is because it is a very extensive tree that practically assigns a node per each instance: It has 3139 nodes and is 34 levels deep. As well known, hyperparameter tuning and pruning the tree can contribute to solving this problem.

Tuning the hyperparameters with a Grid Search Cross Validation could help to improve the prediction. The cross-validation strategy explained in Section 3.2 is maintained. The `GridSearchCV` function from `sklearn` allows passing in a parameter grid which will then test the best parameters. For Decision Trees, the following configurations are tried for the parameter grid:

- **Criterion.** ([gini, entropy]) : It is the probability of new random data being misclassified when hitting a node. Low Gini scores per node are of interest. However, it is impossible to obtain a tree with all pure leaf nodes (i.e. Gini = 0), as this would be clearly over fitting on our training set. Entropy is a similar measure.
- **Max depth.** ([5,10,15,20]): Depth threshold not based on impurity calculations. The tree stops splitting if this limit is reached.
- **Min. samples split** ([2,3,4,5]) : Minimum number of samples for a split to occur.
- **Min. samples leaf** ([1,2,3,4,5]) : Minimum num. of samples required for each node.
- **Max features** ([`auto`, `sqrt`, `log2`, `None`]): It sets the number of features needed for a split. It also helps prevent overfitting. Note that `sqrt` (same as `auto`) and `log2` indicate the square root and logarithm in base 2 of the number of features respectively.

Besides this grid, it also important to specify on which metric should the CV function should refit using the best parameters. As explained in Section 3.2, the Macro F1 score is chosen. This means that the Grid Search method will return the best parameter configuration based on which one yields the highest F1 score. Table 3.2 shows global results with regard to parameters and metrics. Observe that only a very slight improvement the F1 score is obtained, clearly showing the limitation of the Decision Trees. The weighted score is notably increased, but this is only because more instances are predicted to be of Black race. This comes at a cost: a reduction to almost 0 of predicting capacity for the "White category": The number for false positive increases. This is also seen in the Balanced Accuracy score: it remains barely unaltered . However, this model does overfit, most likely due to the CV procedure: Trying it on the train set, returns an F1 score of 0.75, instead of the near perfect score obtained earlier.

Model	Acc.	F1 per class	Macro F1	Weighted F1	Bal Acc.	Depth
DT-default	0.616	[0.72, 0.117, 0.48]	0.439	0.622	0.444	36
DT-best	0.676	[0.788, 0.074, 0.39]	0.418	0.638	0.414	15

Table 3.2: Parameter and metric comparison between Decision Tree models

Parameters for the trees are as follows : It used Gini as the proabilistical criteria, the max depth is 15, the max features is set to the square root, the minimum sample leaves is 3 and the min samples split is equal to 2.

Results from the field of Logics such as the Condercet's Jury Theorem show that combining weak learner and then combining produces a better result than the "Expert Rule" [8]. This is why only trying a Decision Tree as a predictor model is a mistake.

Random Forests are seen as an evolution of the Decision Tree model. They are conceived as an ensemble, which is defined as a collection of multiple ML algorithms that all together can obtain better predictive performance than if used alone. Random Forest trains several trees by splitting the data into subsets. This idea stems from the concept of **bootstrap aggregating** or **bagging**, which allows for a better estimation of predictors with high variance. This means that the different predictions from Decision Trees, which are models with high variance are combined.

Table 3.3 shows an improvement in all metrics for Random Forests.

Model	Accuracy	F1 per class	Macro F1	Weighted F1	Balanced Acc
DT-default	0.616	[0.72, 0.117, 0.48]	0.439	0.622	0.444
RF-default	0.738	[0.83, 0.089, 0.51]	0.471	0.704	0.465

Table 3.3: Comparison between the DT and RF base models

An intermediate step in the Random Forest is to try the same algorithm incorporating class weights. This makes sense, because as stated in section 3.1, the Stop, Question and Frisk Dataset is clearly imbalanced. See results in Table 3.4.

Finally, Stratified Cross Validation is applied with several configuration parameters to find the best Random Forest. Note the high training time for the myriad of possibilities. This is why a subset of possible parameters is taken in order to run the Grid Search for Random Forests. Observe some of the new hyperparameters tried (with respect to Decision Trees): **Number of estimators** is the number of trees in the forest (i.e. [200,400,600]). **Bootstrap** refers to the method for sampling data points(with or without replacement) (i.e. [True, False]). Finally, **Balance** adapts to imbalanced data set, so in this case most likely will be set to True. (to improve performance).

Model	Accuracy	F1 per class	Macro F1	Weighted F1	Balanced Acc
RF-default	0.738	[0.83, 0.089, 0.51]	0.471	0.704	0.465
RF-balance	0.732	[0.826, 0.089, 0.5]	0.472	0.696	0.458
RF-best	0.696	[0.784, 0.192, 0.543]	0.507	0.69	0.50

Table 3.4: Comparison between RF configurations

The best parameters are found to be bootstrap = False, Class Weight = Balanced, Max Depth = 20, Minimum Samples leaf = 4, Minimum Samples split = 2, Number of Estimators = 200.

Finally, note that External Tree Classifiers are tested on the best Random Forest model. Extra Trees add randomization by choosing split-points at random. Hence, the training is faster and more randomized. However, results from the best RF model are not improved, as shown in Table 3.5.

Model	Accuracy	F1 per class	Macro F1	Weighted F1	Balanced Acc
RF-best	0.696	[0.784, 0.192, 0.543]	0.507	0.69	0.50
Extra-best	0.672	[0.77, 0.190, 0.533]	0.498	0.675	0.505

Table 3.5: Random Forest vs. Extra Trees

It is clear that the optimized Random Forest model is the best performing tree-based model with a macro F1-score of 50.7 %. It predicts 'Black' with 78.4 % accuracy, 19.3 % accuracy on 'Other' and 54.3 % on 'White'. The Confusion matrices for this model are shown in **??** and **??** These results are considered accurate, given that they generalize well and predict something as complex as race categories like 'Black' and 'White' well over 50 %. A major advantage of tree-based models is that they are quite transparent in their prediction and keep track of feature importance, thus yielding some insight necessary to solve the main research questions. An overview of the feature importance is shown in Figure 3.3 of the top numerical predictors, averages per race-group are shown in Table 3.6 As top predicting feature, it is clear that stopped Black people are generally younger. The second most important, less obvious predictor is the stop duration minutes showing that Police stops of Black people take significantly less time. The time of the stop, shows that stopping Black people happens later in the day, while White people are stopped earlier. Then some physical attributes are used, like Black people on average being taller and heavier. Figure 6.11 in the Appendix shows that Queens is most probably the best predictor for the 'Other' category with a significant percentage of this groups' stops happening there, which is probably highly correlated with the significant amount of Asian people living in this borough [9]. Finally, in Figure 6.12 in the Appendix the distribution of Kmeans behaviour can be seen per race group. The most significant three differences are in '0' (= 'other' group), '1' (= 'calm' group) and '4' ('cooperative' group). Showing that Black people are in general described by Police Officers as less calm but more often cooperative. The 'Other' group being calm but often less cooperative and the 'White' group somewhere in between.

Table 3.6: Averages per race of significant numerical predictors.

	Black	**Other**	**White**
Reported Age	29.01	30.32	31.31
Stop Duration Minutes	9.60	11.80	11.11
Stop Frisk Time Hour	13.05	12.93	12.39
Suspect Height	5.66	5.62	5.62
Suspect Weight	169.26	163.17	166.00

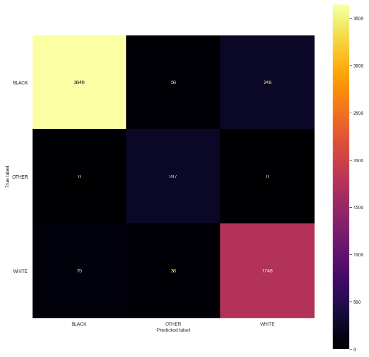


Figure 3.1: Best Random Forest on Training Set

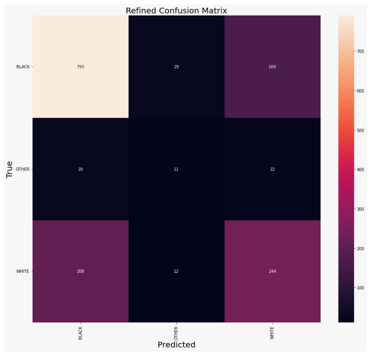


Figure 3.2: Best Random Forest on Validation Set

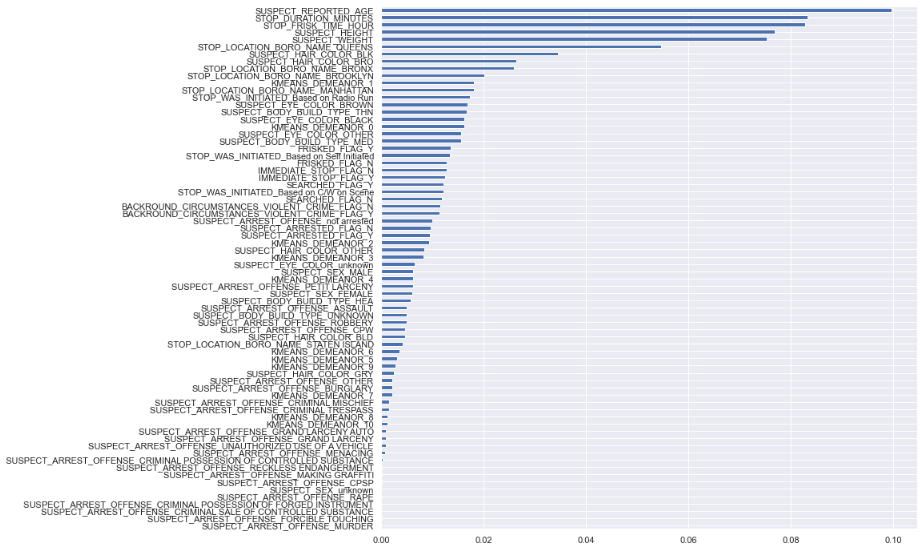


Figure 3.3: Feature Importance in the best RF model

3.4 Multilayer Perceptron

The second model considered is the Multilayer Perceptron. Again, the validation strategy mentioned in Section 3.2 is applied.

As input, the variables from the feature selection (Section 2.2) are used. Because Neural Networks are considered to perform better on standardized numerical variables, a standard scaler is applied on the numerical variables. In terms of categorical variables, two options are explored: both the data set with the original categorical variables and the one with MCA dimensions is tried. Because the neural network can't directly handle categorical data, the categories in the original data frame are One Hot Encoded.

Initially, a baseline using logistic regression is trained for two reasons: First, it requires basically no parameter optimization. Secondly, it can be considered the building block a multi-layer perceptron model as a single-layer, single-neuron model with a logistic activation-function can be considered a logistic regressor. On this initial model a cross-validated macro F1-score of 41.2 % is achieved, with an accuracy of 69.7 %. Looking at the training set confusion matrix, displayed in the Appendix in Figure 6.7, it can be found that the model almost strictly predicts 'Black', to a lesser degree 'white' and only once 'other' (probably such that the macro-F1 average is not 0).

Next, some different neural networks architectures are explored to get a feel for the models' tendencies. Some initial observations are:

- The models build on the one-hot encoded categories performs better than the MCA-based models. This is to be expected as the MCA-based models hold less information (about 50 %

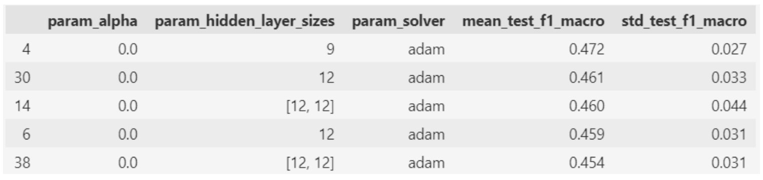
variance) than the original one.

- Building a complex and deep network (fe. three layers of each 15 neurons) overfits on the training set and generalizes poorly. This can be seen in Figure 3.5where the training confusion matrix is exceptionally accurate but the test confusion matrix performs significantly worse. The layer architecture will thus be an important hyperparameter to optimize.
- Cross-validation based model optimization depends heavily on the F1 score chosen (micro or macro). When macro F1 is chosen the model tends to try and predict the 'Others' class too. In contrast, with micro F1, the prediction of 'White' and 'Black' is prioritized.
- The logistic activation function tends to yield the best results.
- The 'lbfgs' leads to quick and often similar imbalanced results while the 'adam' optimizer seems to be better in predicting all three target classes.

Based on these findings, a grid-search with the following configurations is tried:

- **alpha**: ([1e-05, 0.0001, 0.001, 0.01, 0.1, 0.5]) : the regularization term used to regulate over- and underfitting. The higher alpha, the smaller the weights, the more general the solution. Vice versa for lower alpha-values.
- **hidden_layer_sizes**: ([3, 6, 9, 12, [3, 3], [6, 6], [9, 9], [12, 12], [3, 3, 3], [6, 6, 6], [9, 9, 9], [12, 12, 12]]) : These are the different architectures proposed for the model. It represents the amount of layers and how much neurons there are in each layer. The more layers and more neutrons the more expressive the model but also leading to an increased possibilty of overfitting.
- **solver**: (['adam', 'lbfgs']) : The solver is the method used to update the weights based on the training loss. 'adam' is a gradient-descent based method while 'lbfgs' uses quasi-Newton methods.

The grid search took 2:30 hours to complete. The results for the five best performing parameter sets can be found in Figure 3.4 . The best model has the following parameters: ['alpha': $1e-05$, 'hidden_layer_sizes': 9, 'solver': 'adam']. The train- and test confusion matrix corresponding to this model can be found in Figure 3.5.



	param_alpha	param_hidden_layer_sizes	param_solver	mean_test_f1_macro	std_test_f1_macro
4	0.0	9	adam	0.472	0.027
30	0.0	12	adam	0.461	0.033
14	0.0	[12, 12]	adam	0.460	0.044
6	0.0	12	adam	0.459	0.031
38	0.0	[12, 12]	adam	0.454	0.031

Figure 3.4: Top five results of Multilayer Perceptron GridSearch.

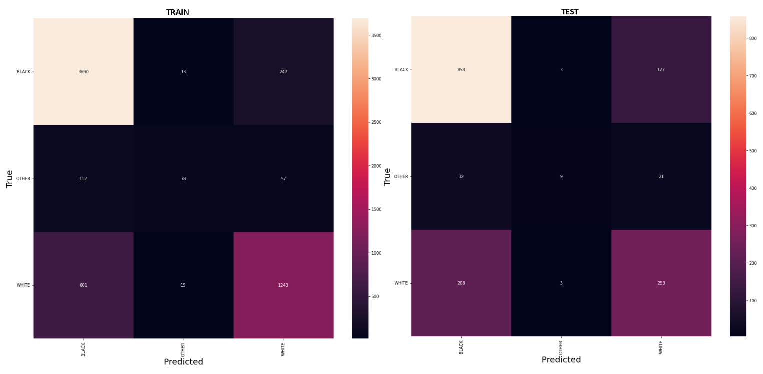


Figure 3.5: Confusion matrix of MLP's best model for training- (left) and validation (right) set.

From the final results it is clear the model overfits less compared to some previous models as it keeps a similar prediction distribution in both train and validation. This is probably the result of the cross-validation strategy of the grid-search. The model yields good results. On the validation set, it yields an average macro F1-score of 54.7 %, the highest score on the validation set of all models considered. It predicts 'Black' and 'White' races with respectively 82.3 % and 58.5 % accuracy, more than acceptable results for predicting something subject to a as much case-to-case variability as 'race'. Moreover, about 23.38 % of the 'Others' instances is predicted correctly, instead of never predicting it (in other models) which is better than nothing. Moreover, when looking at the confusion matrix of the validation set, it can be seen that when the model decided to classify an instance as 'Others', it was the correct prediction more often than not.

Given the optimized MLP model performs the best out of all models of every family considered, it will be used as final model for evaluation on the test set.

3.5 Support Vector Machines

The following family of considered models is Support Vector Machines (SVMs). There are many combinations and parameters to try as they will be unveiled in the following paragraphs.

The problem of this data set is a multiclass classifcation. There are 2 approaches to consider here when using SVM: "one versus one" and "one-vs-the-rest". Since the `sklearn` library is used, note that `SVC()` function uses the first approach and `LinearSVC()` uses the latter.

It is not the objective of this report to provide all the mathematical details behind this complex classification problem. Although a few details are considered necessary in order to justify all the parameter decision when finding choosing the final SVM model.

If R is the number of classes ($R = 3$ for this project), the **one-vs-one approach** constructs $\frac{R(R-1)}{2}$ classifiers. This results in several "SVM equations" to test for each classifier: $y_i(w^T x_i + b) \geq 1 \quad \forall i$ where $y_i \in \{-1, 1\}$ representing classes C_j, C_k. Recall that $w^T x + b = 0$ defines the separating hyperplane. Finally, Majority voting rule is applied to determine the class of the x_i training sample.

The **one-vs-the-rest** approach requires R classifiers. They assign 1 to C_i and -1 to the rest of classes. For example, in the current Data Set the following classifiers would apply: Black against White& Other, White against Black& Other and finally Other against White& Black. Majority voting also applies to classify the data point.

The question is what approach to choose. Observe that for a large $R << \frac{R(R-1)}{2}$, the latter being more computationally expensive. However that is not the case for the since both have $R = 3$. A Linear SVM with OVR approach gives a Macro F1 score of 31.05% and the OVO approach returns an F1 score of 43.1 %. From now on, the different kernels will be OVO-based. Note also that although powerful tools, the SVM problem is of high complexity and therefore . It is estimated that the implementation scales between $O(n_{features} \times n^2_{samples})$ and $O(n_{features} \times n^3_{samples})$ [7].

C is a regularization parameter related to bias-variance trade-off. It measures how much one wants to avoid misclassification in the training set. By default it is set to 1. For large values of C, the hyperplane margin will be smaller and classify more points correctly. A smaller value for C will produce a larger margin hyperplane but could lead to a high misclassification rate. How does the C change the optimization problem formula is detailed in the Appendix. A visual representation of a simple SVM example using just two variables of the data set (age and height) is also provided in the Appendix (Figure 6.13).

The problem with Linear SVM is that it has some limitations with classification. Data sets that cannot be classified well enough with Linear SVM are called non-separable and allow for exploration with non-linear methods. Also called **Kernel methods**, the idea behind them is to move the feature data into a higher dimensional space via what is defined as a **kernel** function. However, the mathematical computations required for this are avoided with the **Kernel trick.**

The kernel functions considered in this project (because they are the more well known) are: **polynomial** (with equation $(\gamma\langle x, x'\rangle + r)^d$), **radial basis function** (with eq. $\exp(-\gamma\|x - x'\|^2)$ and **sigmoid** (with eq. $\tanh(\gamma\langle x, x'\rangle + r)$). Note the introduction of new parameters, which will be discussed in detail for each case. As a rule of thumb, it is good to start with a low degree polynomial or an RBF kernel with a reasonable width is where to start. Before trying any of the different classifiers, this a classification method where scaling the data is a crucial aspect. That is why a pre-processing function is included in the script that scales the numeric columns using `MinMaxScaler()` which sets all the values in a [0,1] scale. Also, the categorical variables are converted to numerical using One hot encoding (exactly the same as for the other models).

The base results (C=1, no other fixed parameters) for the different kernels are shown in 3.7:

Model	Linear	RBF	Polynomial	Sigmoid
Macro F1	0.43	0.358	0.256	0.272

Table 3.7: Macro F1 for base SVM models

Repeated Stratified K Fold CV is applied on the C parameter for a linear kernel. The C parameters tried are in the logarithmic scale $C = \{10^{-2}, 10^{-1}, 1, 10, 100\}$ Because of the complexity, training time is high. As a matter fact, the best parameter is found to be $C = 1$, the one by default. The F1 score is 42% (identical to what was known).

One of the new parameters that appears in non-linear kernels is the γ parameter. This value indicates how much influence a single training example has. A small gamma returns higher values for close points, wheras a higher gamma returns the opposite. In other words, high values of gamma ignore points far away from the decision boundary because interactions returned are lower. This is clearly inferred from the RBF function shown above. It also fairly easy to prove (using the Taylor series of e^x) that these interactions happen in a infinite dimension space.

In polynomial and sigmoid kernel functions, there is the extra r parameter. The d is also another parameter just for the polynomial indicating its degree.

Same cross validation is applied for both RBF, Polynomial and sigmoid, but now with $\gamma = \{0.01, 0.1, 1, 10\}$, $r = \{0, 0.5, 1\}$ and $d = 1$. SVMs incurr Also note that `sklearn` refers to r as `coef0`. Results are summarized in the following list:

- Polynomial Kernel: Best parameters: $C = 0.1$, $\gamma = 10.0$ and $d = 1$, $r = 1$ Macro F1: 43.0%
- Radial Kernel: $C = 1.0$, $\gamma = 0.01$, $r = 0.5$. Macro F1 = 37.7%
- Sigmoid: $C = 0.01$, $\gamma = 0.01$, $r = 1$. Macro F1 = 17.8%

Observe that there is a significant improvement in the performance with the Polynomial Kernel. The Radial Kernel also improves, but is still far away from Polynomial kernel results. The tanh does the worst at this classification task. On the validation set, it classifies all instances as "Black", yielding a useless model.

All the Cross Validations seem to indicate that SVMs cannot go higher than 43% in Macro F1. Recall that SVM is a distance-based model, meaning that the goal is to find hyperplanes that minimize distance between different classes so there is a limit in the optimization in this problem due to the nature of the data set. It also informing to take a look at the Confusion Matrices shown in Figures 3.6 and 3.7

The classification problem is very complex in its nature, and it is clear that Regression Trees and MLPs do a much better task. This is covered in the conclusion (cf. Section 4.)

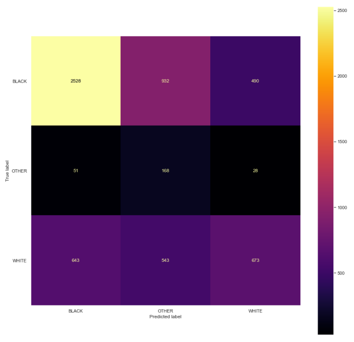


Figure 3.6: Polynomial Kernel on Training Set

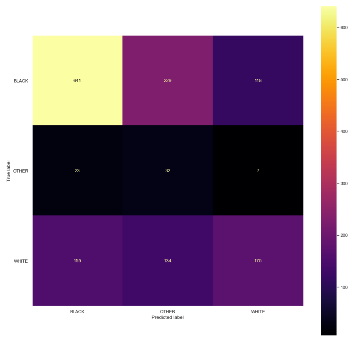


Figure 3.7: Polynomial Kernel on Validation Set

3.6 Final Model

As mentioned earlier, the final model is chosen based on the best average macro F1-score on the validation set. From the three families considered, the optimized MLP model performed the best, with an macro F1-score of 54.7 % on the validation set. This is an improvement on the optimized

Random Forest with an F1-score of 50.7 %. The Support Vector Machine however performed significantly worse with an F1-score of 43.5 %. Given that neural networks are some of the most powerful and expressive models available, it is no surprise it performs well. Moreover, the large amount of parameters (layer size, neuron amount, regularization, etc.) allow for controlling overfitting and dealing with the imbalanced target distribution without having to use undersampling. It is expected that these scores will generalize well to the test set as appropriate methods were used to minimize data leakage and overfitting in model training and selection.

Now, the MLP model is applied on the test set. Note that in doing this, some variable alignment has to be done between the model and the test set. More details on this can be found in the Appendix in Section 6.2.2. **This final evaluation on the test set yields an average macro F1-score of 53.4 %. It predicts 'Black' with 81.2 % accuracy, 'White' with 55.6 % accuracy and 'Other' with 23.0 % accuracy**. These results are very similar to the validation scores (average macro F1 of 54.7 %), thus proving that the model generalized well (low generalization error) and model training can be considered successful. The final confusion matrix can be found in Figure 3.8. Note that MLP's are generally considered 'black boxes' and tracing back which variables contributed most to prediction is difficult. For interpretability purposes, in Section 4, a Random Forest model is used to find which variables play an important role in prediction.

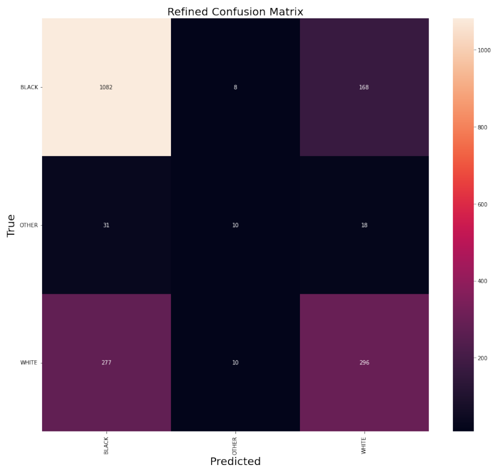


Figure 3.8: Confusion Matrix for predictions of final model on test set.

4 Conclusions

4.1 Scientific

The goal is to predict the suspect race of police stops in New York, hence possibly gaining more insight into the racial profiling and treatment differences that have been reported widely in the United States. First, the data is cleaned in an extensive exploration, pre-processing and feature selection process. These steps reveal already some insight on the topic at hand:

- A disproportional amount of Black people are stopped compared to other race groups.
- Significantly many factors are in play, distinguishing the stop and frisk of a Black person from that of a White person or from other backgrounds. Obvious predictors are neighborhood, height, weight, etc. But some are considered rather unexpected like the time of the stop, the duration of the stop and the behaviour of the suspect.

Secondly, three candidate model families are tried and optimized in order to build a multi-class classifier: Tree-based methods, Multilayer Perceptrons (MLP) and Kernel-based methods. Of these, the MLP yielded the best results, with an average macro-F1 score of 53.4 % on the test set. The Neural Network has a single layer with 9 neurons, a logistic activation function, an optimized stochastic gradient descent mechanism for weight updating and a regulator alpha value of $1e-05$. For a stop, it predicts if it was a Black person with 81.7 % accuracy, a White person with 55.6 % accuracy and a person from different race with 23.0 % accuracy. On the validation set, it was found that an optimized Random Forest (RF) model has comparable results to the MLP, performing only slightly worse. Compared to the MLP, the advantage of tree-based models is that these are more interpretable. Therefore, for insight into the model predictors, RF parameter importance is used. The following conclusions can be made:

- Black people stopped by the police are generally younger, which makes for the best predictor of the model.
- Stops of Black people are generally shorter and happen later in the day. This might be because of high frequency, casual unnecessary stops of Black people at night.
- Most of the 'Others' group are predicted by the fact of the stop happening in Queens.
- The police describes the behaviour of Black people stopped often as not calm or cooperative, while the 'Other' group often gets described as calm or non-cooperative. White people descriptions fall somewhere in between.

4.2 Personal

Building this project for the Machine Learning Course of the UPC has been an interesting process. Mostly, the fact that the data captures a tangible real-word process of personal interest, provided motivation to get insightful and credible results. At the same time, the authenticity of the data, required complex pre-processing tasks and continuous revision of the pipeline. The final results are however considered worth the effort put into the project.

5 Limitations and Outlook

Compared to some other machine learning tasks, a test set macro F1-score of 53.4 % might not be considered very accurate. However, note that predicting the race of a suspect based on manually filled reports, is subject to a lot of case-to-case variation and inter-personal differences. It is reasonable to assume no model can accommodate for this variance. However, more thorough NLP analysis and latent variable construction might improve the models slightly. Moreover, note that some serious bias might be present in the data set too, as it are the police officers themselves filling in the forms, knowing that it might be used for analysis like this. Therefore, to further gain insight in New York Police-work, this analysis would have to be repeated over multiple years while critically assessing the content of the forms and the way they are filled in.

References

[1] T. Mrozla, "New york police department stop, question, and frisk program: Experiences across race and gender," 2014.

[2] Y. Badr and R. Sharma, "Data transparency and fairness analysis of the nypd stop-and-frisk program," *J. Data and Information Quality*, vol. 14, feb 2022.

[3] S. Goel, J. M. Rao, and R. Shroff, "Precinct or prejudice? understanding racial disparities in new york city's stop-and-frisk policy," *The Annals of Applied Statistics*, vol. 10, no. 1, pp. 365–394, 2016.

[4] A. G. Bluman, *Elementary Statistics. A Step by Step approach. 8th Edition.* 2009.

[5] A. V. Frane, "Are Per-Family Type I Error Rates Relevant in Social and Behavioral Science?," *Journal of Applied Statistical Methods*, vol. 14, no. 1, pp. 12–23, 2015.

[6] M. Kendall, "A New Measure of Rank Correlation," *Biometrika*, vol. 30, no. 1, pp. 81–93, 1938.

[7] S. learn developers, "Metrics and scoring: quantifying the quality of predictions," 2022.

[8] M. Martín, "Machine learning lecture notes," 2022.

[9] U. C. Bureau, "Population statistics on queens, new york," 2022.

6 Appendix

6.1 Figures and Tables

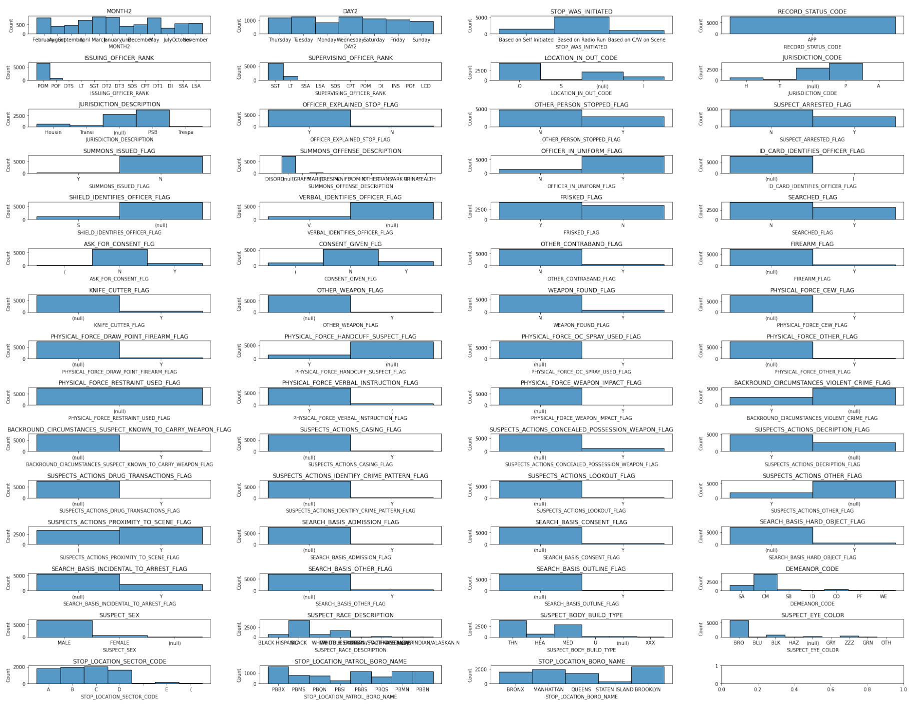


Figure 6.1: Histogram plots of categorical variables in the unprocessed training set.

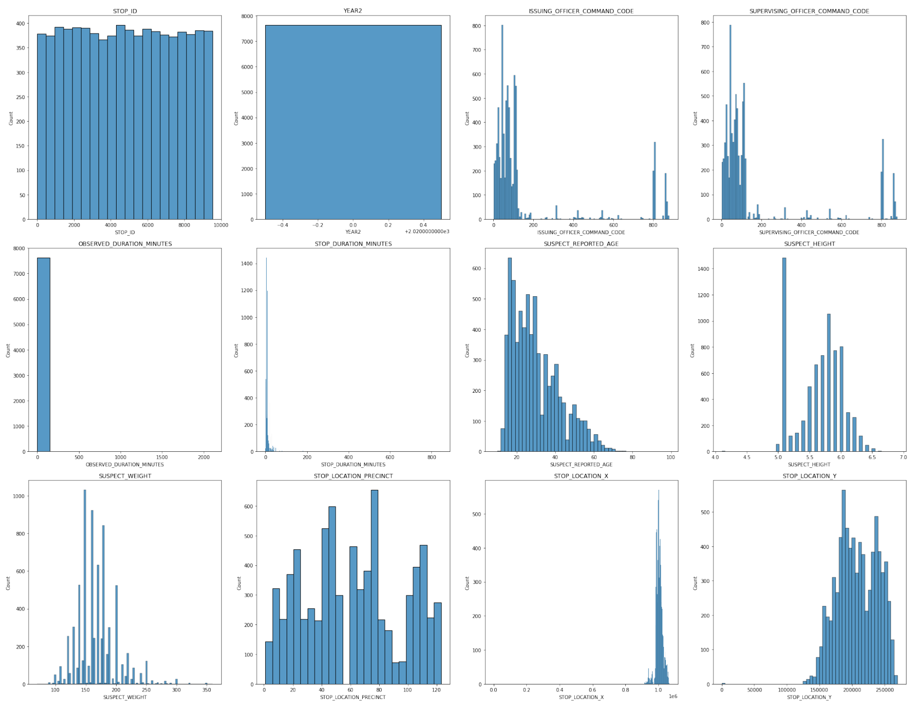


Figure 6.2: Histogram plots of numerical variables in the unprocessed training set.

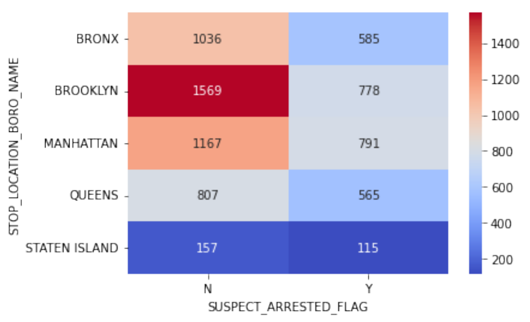


Figure 6.3: Cross tab showing the counts of the binary arrests varibale in relation with the stop location.

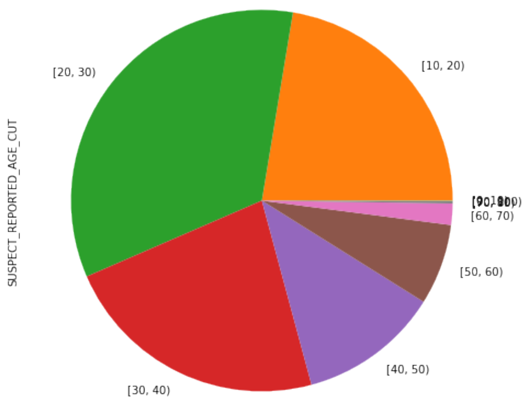


Figure 6.4: Pie chart displaying the age distribution of the data set.

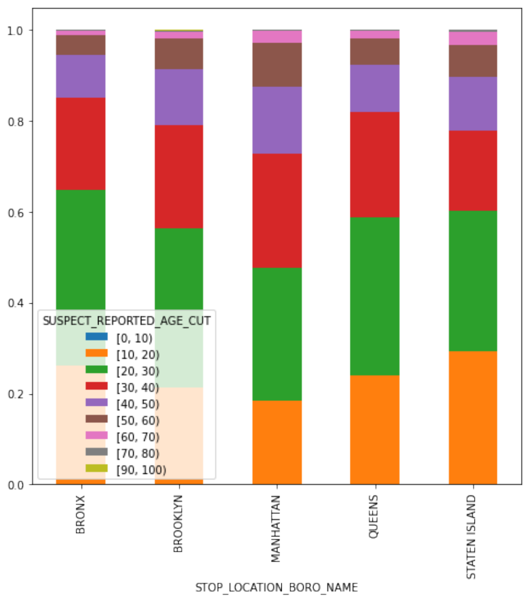


Figure 6.5: Cross tab displaying the relation between age and NY borough in the data set.

Table 6.1: Mapping of semantic meaning of each group as a result of NLP Kmeans on 'suspect demeanor description'.

Cluster Label	**Semantic Meaning**
'0'	rest group
'1'	calm group
'2'	nervous group
'3'	normal group
'4'	cooperative group
'5'	agitated group
'6'	upset group
'7'	compliant group
'8'	angry group
'9'	confused group
'10'	calm & cooperative group

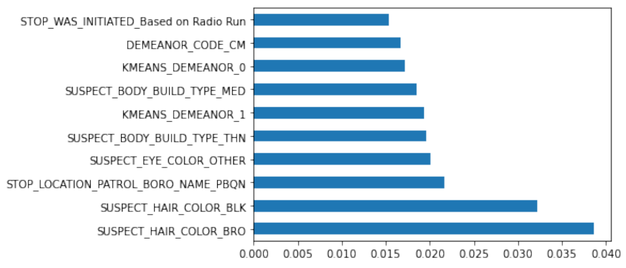


Figure 6.6: Best 10 features using Extra Trees

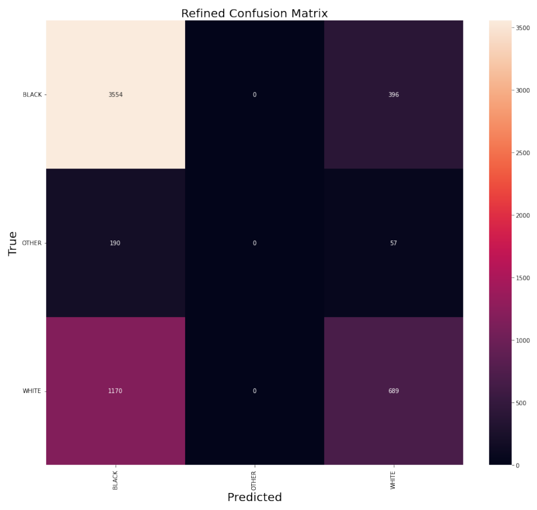


Figure 6.7: Confusion matrix on training set of Logistic Regression Model.

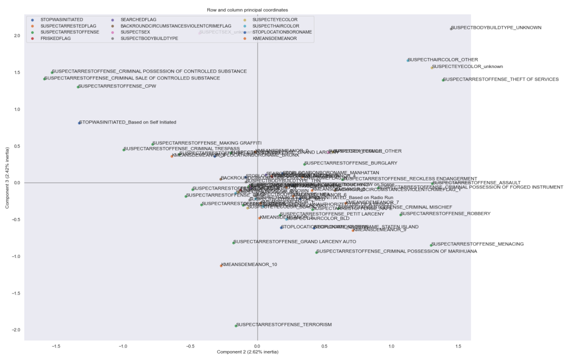


Figure 6.8: MCA Dim 2 and 3

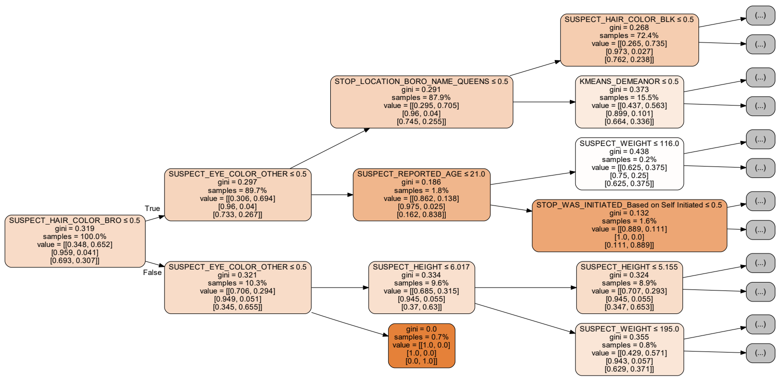


Figure 6.9: Default Decision Tree model with depth=3

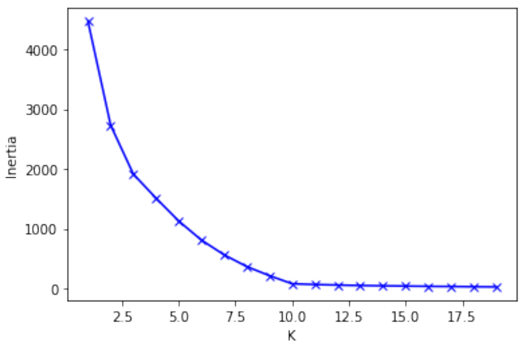
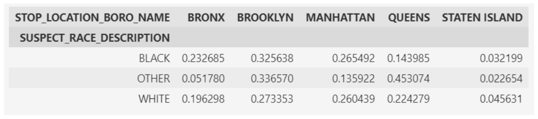
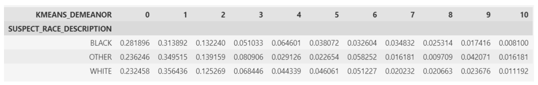


Figure 6.10: Inertia loss for increasing k in k-means applied on demeanor descriptions.



STOP_LOCATION_BORO_NAME	BRONX	BROOKLYN	MANHATTAN	QUEENS	STATEN ISLAND
SUSPECT_RACE_DESCRIPTION					
BLACK	0.232685	0.325638	0.265492	0.143985	0.032199
OTHER	0.051780	0.336570	0.135922	0.453074	0.022654
WHITE	0.196298	0.273353	0.260439	0.224279	0.045631

Figure 6.11: Percentual distribution of borough stop location per race group.



KMEANS_DEMEANOR	0	1	2	3	4	5	6	7	8	9	10
SUSPECT_RACE_DESCRIPTION											
BLACK	0.281896	0.313892	0.132240	0.051033	0.064601	0.038072	0.032604	0.034832	0.025314	0.017416	0.008100
OTHER	0.236246	0.349515	0.139159	0.080906	0.029126	0.022654	0.058252	0.016181	0.009709	0.042071	0.016181
WHITE	0.232458	0.356436	0.125269	0.068446	0.044339	0.046061	0.051227	0.020232	0.020663	0.023676	0.011192

Figure 6.12: Percentual distribution of kmeans behaviour groups per race group.

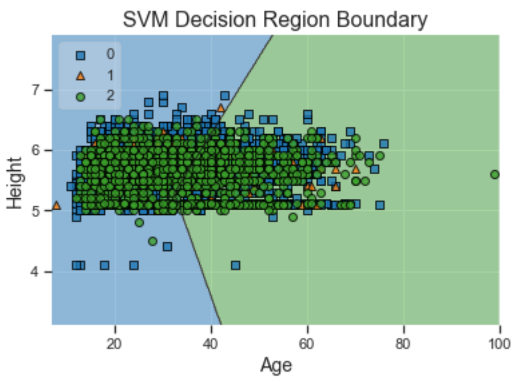


Figure 6.13: Simple SVM example using 2 variables of the Data Set. Classes indicate Race predicted as Black, Other and White respectively

6.2 Additional Text

6.2.1 SVM Classification Problem

If the error in classification is $\xi \geq 0$, that means that the SVM optimization problem is now:

$$\text{Minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to } y_i(w^T x_i + b) \geq 1 - \xi_i$$

6.2.2 Model-test alignment

When variables are one-hot encoded, a new variable is made per possible category level. This can pose some problems. Take for instance, a rare category level appears in the training set and is this part of the trained model but does not appear in the test set. The test set would need to have this variable too, set to value 0 everywhere of course. To do this, an empty data frame containing only the columns in the one-hot encoded training set is stored in `.csv` format. This is then loaded in at when applying the model on the test set and aligned to the test set using pandas' `align` function. Note that no data leakage is present, as all values are filled with zeros.