UNIVERSITÀ DI PISA

DEPARTMENT OF COMPUTER SCIENCE

MASTER'S DEGREE IN DATA SCIENCE & BUSINESS INFORMATICS

DISTRIBUTED DATA ANALYSIS & MINING

ACADEMIC YEAR 2020/2021

Distributed analysis of homicides' solvability and delinquents' characteristics in the USA

Authors:
Tommaso CAVALIERI [597707]
Federica GUIDUCCI [600310]
Valentina OLIVOTTO [600009]

January 11, 2021

Contents

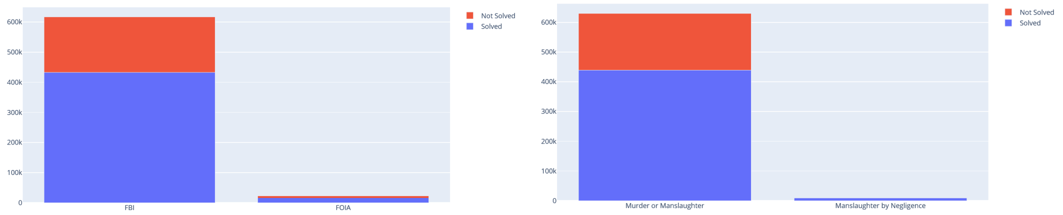
1. Data Understanding & Preparation

The project described in this report consists in the analysis of a dataset of homicides, both solved and unsolved, committed on the United States' territory between the years 1980 and 2014. Such records were derived from the FBI's Supplementary Homicide Report and Freedom of Information Act data. All the data mining and analysis performed for completing this project, was implemented thanks to a distributed computation, enabled by the usage of the **Apache Spark™** framework, which was largely exploited to complete the tasks that will be described in the following paragraphs. The complete list of the Python libraries utilized for this project is reported in the Appendix A. The original dataset contained 638,454 records, each one of them representing a murder, and 24 attributes, whose meaning was deducted from context. After a preliminary analysis, we decided to drop the following four columns: *Record ID*, *Incident*, *Victim Count* and *Perpetrator Count*. The first one was not significant because it represented an identification number, while the other three attributes did not have any clear meaning or usefulness because of the lack of information provided on Kaggle, the platform where the dataset was found and downloaded from. In Table 1.1 are reported all the original attributes and their domain; please notice that originally the type of *Perpetrator Age* was set to *string*, but then we cast it as an *integer*.

Attribute type	**Name**	**Domain**
String	Agency Code	{CA03902, FL05600, 1201 MORE}
	Agency Name	{ Santa Paula, Worcester, 9214 MORE}
	Agency Type	{ Sheriff, Special Police, 5 MORE}
	City	{East Feliciana, Aitkin, 1780 MORE}
	State	{ Pennsylvania, Connecticut, 49 MORE}
	Month	{January, February, ..., December}
	Crime Type	{ Manslaughter by Negligence, Murder or Manslaughter}
	Crime Solved	{ Yes, No }
	Victim Sex	{ Male, Female }
	Victim Race	{ Native American/Alaska Native, White, 3 MORE}
	Victim Ethnicity	{ Not Hispanic, Hispanic, Unknown }
	Perpetrator Sex	{ Male, Female }
	Perpetrator Age	{ 0, 1, ..., 99}
	Perpetrator Race	{ Native American/Alaska Native, White, 3 MORE }
	Perpetrator Ethnicity	{ Not Hispanic, Hispanic, Unknown }
	Relationship	{ Stranger, Father, 26 MORE}
	Weapon	{ Handgun, Suffocation, 14 MORE }
	Record Source	{ FBI, FOIA }
Integer	Year	{1980, 1981, ..., 2014}
	Victim Age	{ 0, 1, ..., 998 }

Table 1.1: Conserved attributes from the original dataset

During this first phase we discovered that there were different cities sharing the same name in different States so, in order to avoid confusion between them, we created a new column called *City_State* in which *City* and *State* were merged together. With the aim of removing redundancies, we later decided to drop the column *City* and keep the attribute *State* because of its higher granularity, which revealed to be very useful in the following stages. Moreover, we decided to delete *Agency Name* because it was not significant for the purpose of our analysis and we kept only *Agency Code*, which was unambiguous and allowed to identify each agency uniquely. Unfortunately, we found out that the features *Crime Type* and *Record Source* were, as shown in Figure 1.1, heavily unbalanced and hence we discarded both of them. In the course of the analysis some semantic errors were also discovered, for example *Victim Age* took value 998, which was clearly a way of coding missing values, while the number of records with *Victim Age* = 99 and *Perpetrator Age* = 99 seemed to be suspiciously higher then the ones with the age equal to a near value such as 97 or 98, therefore it was considered as a misspell of the 998 code. The records presenting *Perpetrator Age* < 10 were also considered as wrongly collected data and thus filtered, since it seemed unrealistic to us having murderers that are less than 10-years-old. The dataset did not contain any clear NaN, but we found some missing values in various columns which presented as value the string *Unknown*. Therefore, we removed all the records having such value in one of the Victim's attributes. Moreover, we deleted the records representing a

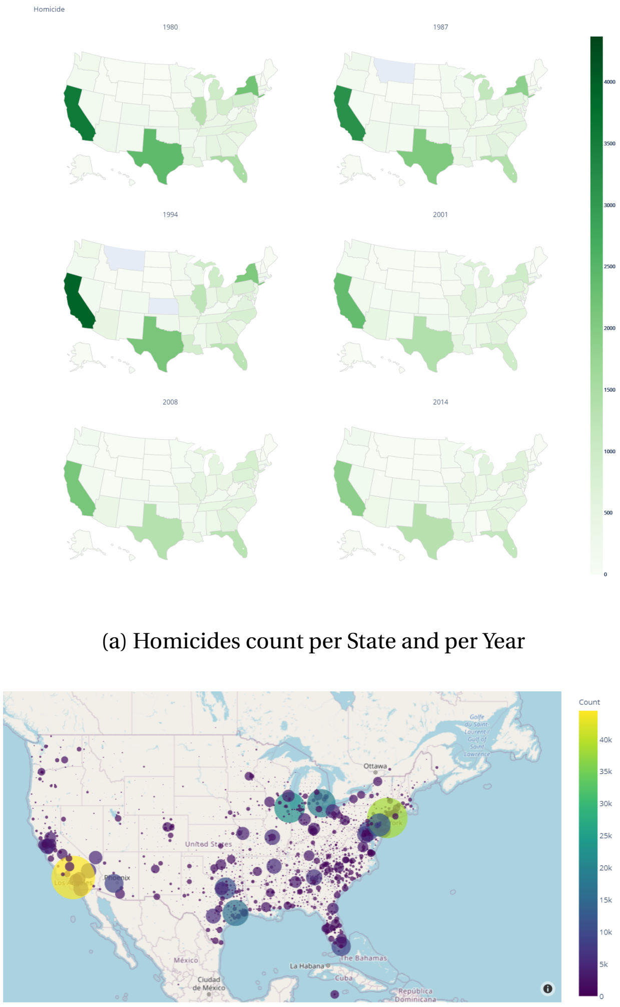


(a) Crime Source

(b) Crime type

Figure 1.1: Unbalanced attributes

solved crime but with one of the Perpetrator's attributes set to *Unknown*, because they appeared to be nonsense and represented just a very tiny fraction of the data. Overall the data cleaning process brought to the deletion of 20,293 records, corresponding to 3.1% of the original dataset. Eventually we implemented some variables' transformations in order to correct some syntactic errors in the column *Relationship* (e.g. *Employer* → *Employee*), even though such variable did not prove to be useful during the following steps of the project. For some attributes we also decided to plot some simple visualizations that could help us get a deeper insight of the data. For example, we exploited geographical attributes as *State* and *City_State* in order to generate a map of the USA displaying the number of homicides per region, as shown in Figure 1.2. Thanks to these plots, we were able to observe the territorial distribution of criminality, noticing that every year 3 states emerged as leaders in terms of homicides committed on their soil: California, Texas and New York.



(a) Homicides count per State and per Year

(b) Homicides count per City

Figure 1.2: Map plot

A temporal view of the distribution of crimes between solved and unsolved can be seen in Figure 1.3, where it is possible to appreciate that even though for some years the count of homicides is higher than others, the trend does not seem to be neither increasing nor decreasing.

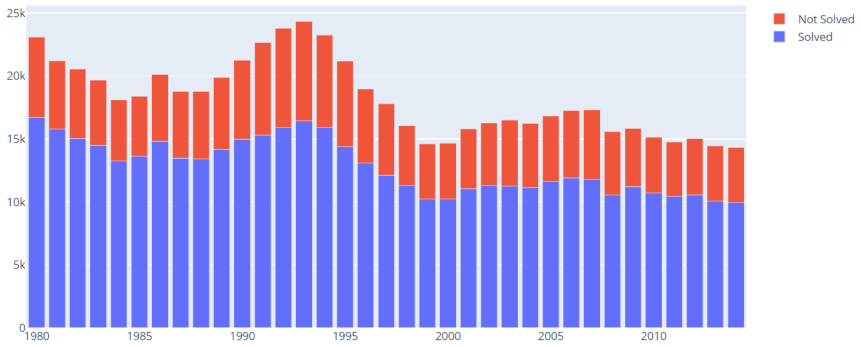
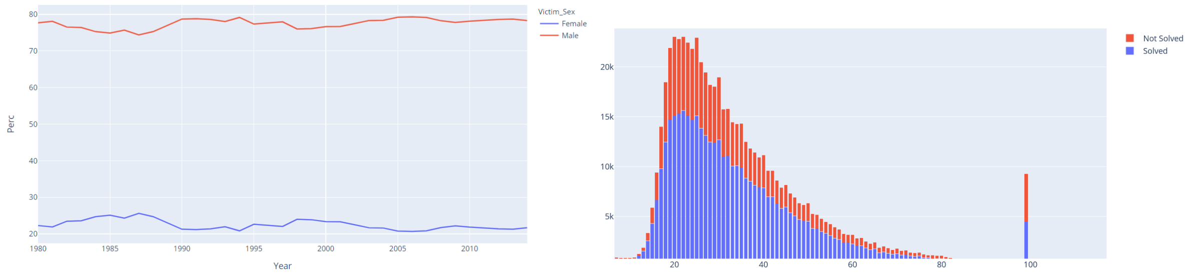


Figure 1.3: Distribution of Crime Solved per Year

The following figures show some peculiar properties of our dataset: in Figure 1.4 we can observe that the majority of the victims were males and that both trends, male's and female's, remained constant over the years. The distribution of the attribute Victim Age is shown in Figure 1.4b, where we can notice the outlying number of records with *Victim Age* = 99 present in the data before the cleaning process.



(a) Victim Sex per Year

(b) Victim Age

Figure 1.4: Victim Attributes

2. Classification Crime Solved

The data cleaning process previously described was completed focusing on one main objective: creating a classification model whose goal was to identify whether a crime had been solved or not, basing such decision on the available attributes. Such features were accurately selected in order to obtain results as little polluted as possible. Some of them, such as *Relationship* or the ones referring to the perpetrator (*Perpetrator Age, Perpetrator Race* etc.) were not exploited in this task, since they would have influenced too much the results. In fact, the model that was built including them ended up giving a classification with 99% accuracy, which, besides the high value, was not satisfactory at all since it exploited a trivial reasoning: it assigned the value *NO* to the target label of every record having in any of those variables the value *Unknown* (which *de facto* indicated a missing value) and *YES* to all the others. Probably such behaviour was due to the fact that for almost every record in the original dataset with *Crime Solved* = *NO*, such attributes were *Unknown*. Various trials were implemented to identify the best set of attributes to be chosen, after which the following decisions were taken:

- the variable *City_State* was kept only in the classifications by State presented in Section 2.2, while in the one applied on the total dataset the variable *State* was preferred, since *City_State* seemed uselessly detailed, while keeping both would have been redundant;
- *Victim Ethnicity* was removed since it did not add any information to the classification and moreover it either took the value *Hispanic* or *Unknown*, thus appearing to be incomplete or wrongly-defined;
- *Agency Code* was removed as it was an identification number and therefore not useful for the classification.

Moreover, after some further experiments it was decided to discretize the variable *Victim Age* into 5-years-classes. Therefore, the attributes fed to the built models were the following: *Agency Type, State, Month, Year, Victim Sex, Victim Race, Weapon* and *Victim Age*. As previously anticipated, beyond the classification on the whole dataset, three different subsamplings were derived by taking only the records from a specific State. Such technique was applied for the three States with the highest number of homicides committed discovered in the previous section: New York, California and Texas. These datasets (the total and the subsampled) were then used to train four different models: the logistic regression, the decision tree, the random forest and the multilayer linear perceptron, whose performances will be assessed in the following paragraphs. Please notice that both for decision trees and random forests a cross-validation technique was implemented to perform the hyperparameters tuning. Moreover, only the logistic model could be fed directly the data as an **rdd**, while MLP, DT and RF needed some preprocessing: all the attributes were first converted to indexes and then merged together in a vector which was used to train the models.

2.1. Complete dataset

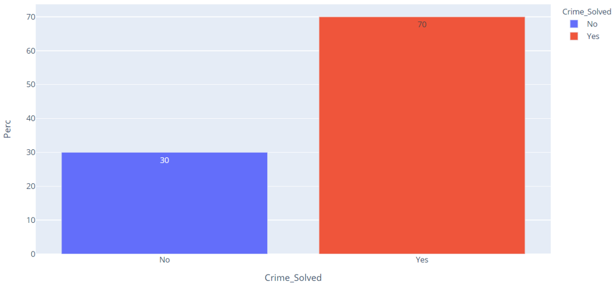


Figure 2.1: Distribution Crime Solved

Initially, the classification models were applied to the original dataset, whose target label's distribution was a little unbalanced (70-30 as shown in Figure 2.1); unfortunately the results obtained were trivial (accuracy = 70%) and therefore the dataset was balanced with a random undersampling of the majority class and the implementation of the models was repeated. This choice was due to the computation cost, which would have been to high for the application of an oversampling technique to the minority class. In order to built the DT and RF, an hyperparameter tuning was necessary: the best performing decision tree had a depth equal to 5 and used entropy as impurity measure, while the selected random forest was composed of 40 trees and exploited the Gini index as impurity. Furthermore, we checked whether the Random Forest model might end up overfitting by plotting the accuracy for each number of trees up to 120 for both impurity measures, as shown in Figure 2.2.

Model	**AUC**	**Accuracy**	**Precision [1 - 0]**	**Recall [1 - 0]**	**F1 score**
Multilayer Perceptron *(layers = [8, 16, 12, 6, 4, 2])*	0.5	0.5004	0.2504 (weighted)	0.5004 (weighted)	0.3337
Logistic	0.5810	0.5808	[0.5691 - 0.5966]	[0.6571 - 0.5048]	0.541943
Decision Tree *(NumNodes = 1195)*	0.6169	0.6165	[0.6466 - 0.5863]	[0.6109 - 0.6229]	0.6282
Random Forest *(NumTrees = 40)*	0.6586	0.6582	[0.6345 - 0.6819]	[0.6671 - 0.6501]	0.6504

Table 2.1: Classification evaluation

As can be appreciated in Table 2.1, the MLP gave the worst and most trivial results, classifying every crime as solved and thus resulting in a 50% accuracy and a 0.5 area under the curve. For this reason, the **ml** evaluation functions had to be applied for this model, while for the others the **mlib** library was preferred because of its better approximation of the results; this explains the different format of the precision and recall measures for the MLP. Various tests were implemented with different configurations of the neural network, changing both the number and the dimensions of each layer, but failing to improve the results: the MLP was thereby excluded from the implemented models in the following phases. The other classifiers behaved as expected: the Random Forest outperformed the Decision Tree and they both outperformed the Logistic Regression. For the classifiers that allowed it, the feature importance of each attribute in the model was also calculated and the results are shown in Figure

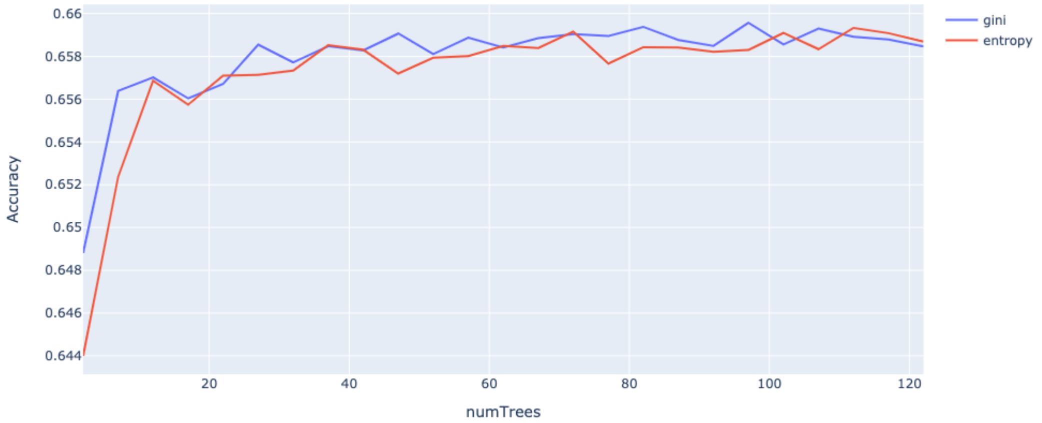
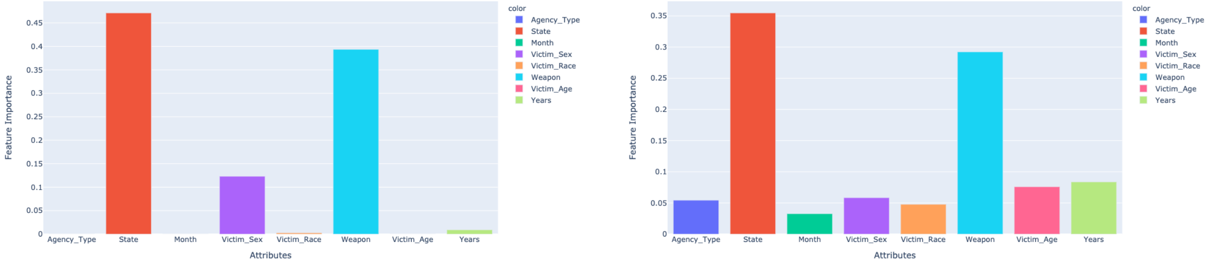


Figure 2.2: RF accuracy per number of trees

2.3. It can be seen how the variables *State* and *Weapon* played a very important role in the classification, thus it seemed interesting to get an insight of the States with the most homicides, which will be presented in the following paragraph. Moreover, it was observed that all the variables that seemed useless in terms of feature importance for the DT, gained more relevance in the more complex model of the RF.



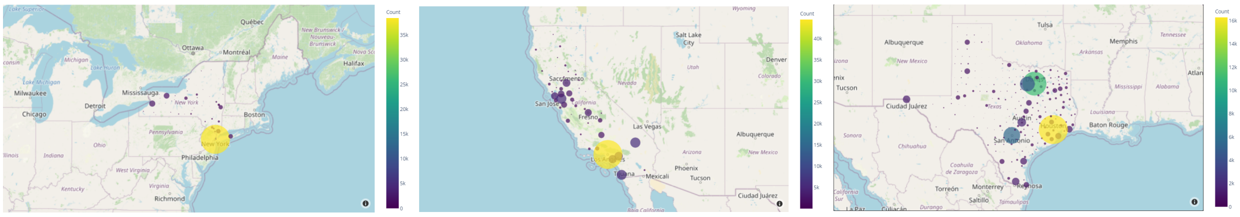
(a) Decision Tree

(b) Random Forest

Figure 2.3: Feature importance

2.2. Main States: CA, TX, NY

For this task, the same attributes of the previous one were exploited, with the exception of the variable *State*, whose usage would have been pretty meaningless in this case because of the implemented subsamplings. It was therefore preferred to include in these models the attribute *City_State*, originally excluded. Another difference with regard to the models built on the whole dataset was the balancing method. In fact, this time the datasets were way smaller (TX: 87,528, NY: 46,601, CA: 130,873) and therefore an oversampling of the unsolved crimes could be applied. Please note that the State of New York already had a balanced distribution and thus did not need this manipulation. In Figure 2.4 is plotted a focus on the three areas. The results obtained with the different models are shown in Tables 2.2, 2.3 and 2.4. Please notice that even though an hyperparameter tuning was implemented beforehand, it gave almost always the same result, selecting Gini as the impurity measure, 1 as the lower bound for the number on instances per node and 10 as the maximum depth of the tree. A small difference was instead obtained for the number of trees of the Random Forest, which is reported in the tables. The models showed a similar behaviour to the one they had in the previous task: the RF was confirmed to be the best classifier, outperforming both the DT and the Logistic regression. It can be noticed how the Texas subsampling was the only one that actually brought to an improvement in terms of performances. For what concerns the feature importance, whose results are shown in Figure 2.5 in this applications the variable *City_State* substituted the variable *State* not only semantically, but also as one of the most important attributes in the classification, along with *Weapon*, just as in the previous models. Furthermore, the variable *Years* showed an increasing importance in this task; minor but yet not ignorable information in the classification process was also added by *Month* and *Victim Age*.

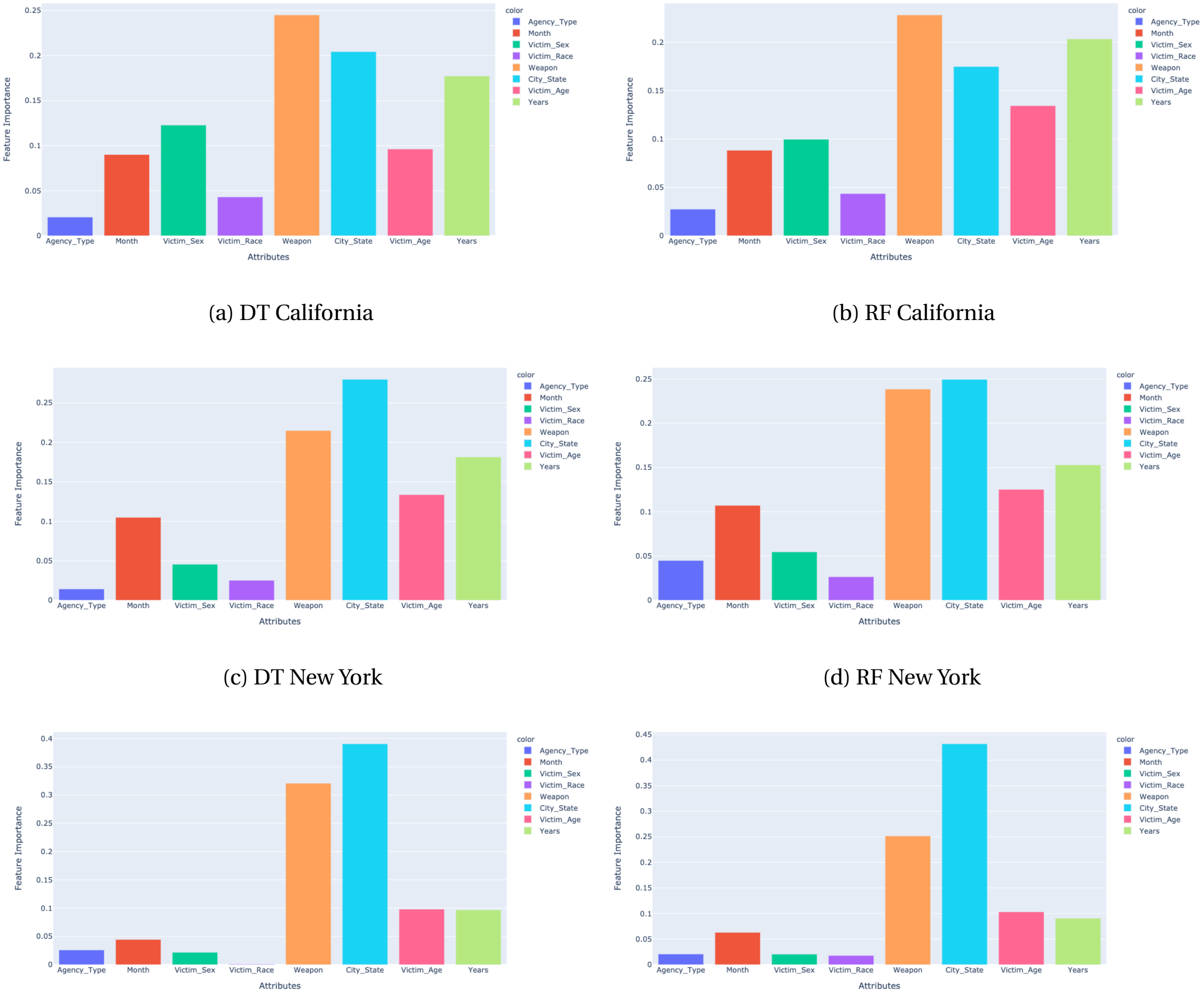


(a) New York

(b) California

(c) Texas

Figure 2.4: Main States' crime distribution



(a) DT California

(b) RF California

(c) DT New York

(d) RF New York

(e) DT Texas

(f) RF Texas

Figure 2.5: Feature Importance

Model	**AUC**	**Accuracy**	**Precision [1 - 0]**	**Recall [1 - 0]**	**F1 score**
Logistic	0.5486	0.5655	[0.5637 - 0.5661]	[0.3000 - 0.7973]	0.3916
Decision Tree *(NumNodes = 1195)*	0.6253	0.6275	[0.5341 - 0.7084]	[0.6136 - 0.6369]	0.5711
Random Forest *(NumTrees = 70)*	0.6404	0.6426	[0.5917 - 0.6867]	[0.6208 - 0.6599]	0.6059

Table 2.2: California

Model	**AUC**	**Accuracy**	**Precision [1 - 0]**	**Recall [1 - 0]**	**F1 score**
Logistic	0.5708	0.5733	[0.5422 - 0.5997]	[0.5350 - 0.6067]	0.5386
Decision Tree *(NumNodes = 931)*	0.6120	0.6099	[0.6454 - 0.5792]	[0.5703 - 0.6538]	0.6056
Random Forest *(NumTrees = 70)*	0.6349	0.6345	[0.6513 - 0.6199]	[0.5972 - 0.6726]	0.6231

Table 2.3: New York

Model	**AUC**	**Accuracy**	**Precision [1 - 0]**	**Recall [1 - 0]**	**F1 score**
Logistic	0.5007	0.5219	[0.4726 - 0.5304]	[0.1472 - 0.8542]	0.2245
Decision Tree *(NumNodes = 367)*	0.6607	0.6619	[0.6096 - 0.7085]	[0.6503 - 0.6711]	0.6293
Random Forest *(NumTrees = 100)*	0.6940	0.6924	[0.6015 - 0.7733]	[0.7024 - 0.6857]	0.6481

Table 2.4: Texas

3. Classification Perpetrator Attributes

In this final section we will describe the last implemented assignment, where we focused on data that could be useful to the police or any other investigation force in order to solve more crimes by creating an accurate identikit of the perpetrator. In fact, the attributes in our dataset that seemed to be more helpful for the investigations were the ones referring to the murderer, in particular his/her age range and sex. Therefore, we implemented two classification models (DT and RF) in order to predict such attributes, basing the decision on the other attributes available in the original dataset. The task could thus be subdivided into three sub-problems:

- Binary classification with Perpetrator Sex as target attribute
- Multiclass Classification with Perpetrator Age as target attribute, discretized every 10 years (9 classes)
- Multiclass Classification with Perpetrator Age as target attribute, discretized every 25 years (4 classes)

In each one of them we started from the original dataset and removed *Crime Solved*, *City_State* and *Agency Code* since they were useless for the purpose of this stage and *Relationship* because it was a perpetrator's attribute and hence could have influenced the classification. As anticipated, we decided to report only the results obtained with DT and RF at this point because in the previous tasks they had given the best results, while the other implemented classifiers failed to perform as expected. In fact, even though we tried to implement also the MLP or the Logistic Regression, the results were unsatisfactory and useless, so we decided to exclude them from the report.

3.1. Perpetrator Sex Classification

In order to solve this problem we removed all the records having *Perpetrator Sex = Unknown* since, as previously mentioned, such records represented missing values and thus did not have any relevance for our goal. Furthermore, as we can see in Figure 3.1, *Perpetrator Sex*'s label distribution was very unbalanced between males and females, so we considered appropriate to balance the classes by applying an undersampling to the majority class.

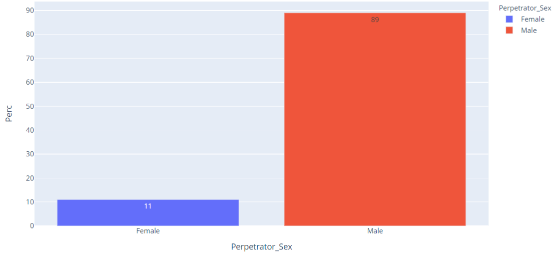


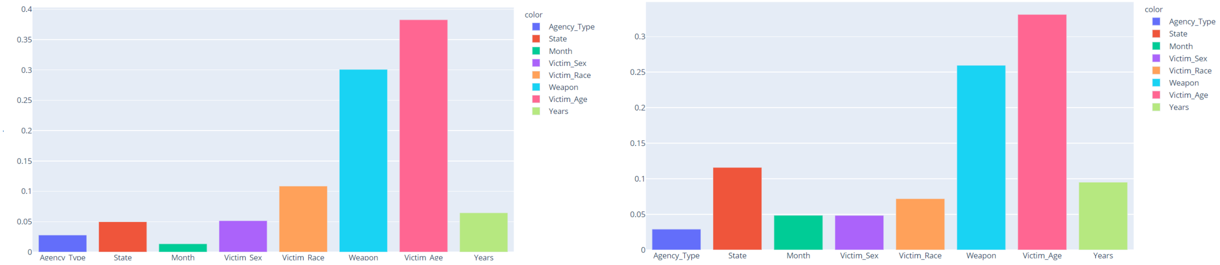
Figure 3.1: Perpetrator Sex distribution

Also in this case we applied a cross-validaton technique in order to asses the best values for the model's parameters. Both classifiers, DT and RF, performed pretty well, giving satisfying results, as shown in Table 3.1.

Model	**Accuracy**	**Precision [1-0]**	**Recall [1 - 0]**	**F1 score**
Decision Tree *(maxDepth = 10, impurity = 'gini')*	0.6826	[0.6672 - 0.6984]	[0.6503 - 0.6711]	0.6748
Random Forest *(NumTrees = 120, impurity = 'entropy')*	0.6860	[0.6451 - 0.7278]	[0.7074 - 0.6677]	0.6759

Table 3.1: Perpetrator Sex Classification

Since both classifiers allowed to compute the feature importance, such information was also derived. As shown in Figure 3.2 the most important features in the classifications were *Victim Age* and *Weapon*, even though every attribute seemed to have its role in determining the perpetrator's gender.



(a) Decision Tree

(b) Random Forest

Figure 3.2: Feature importance per Perpetrator Sex

3.2. Perpetrator Age Classification (9 classes)

In this sub-task we splitted the attribute Perpetrator Age into 9 classes, each of them covering a 10 years span. After such division, we noticed that the distribution between such classes was very unbalanced, as shown in Figure 3.3, so oversampling and undersampling techniques were applied where needed to restore the balance.

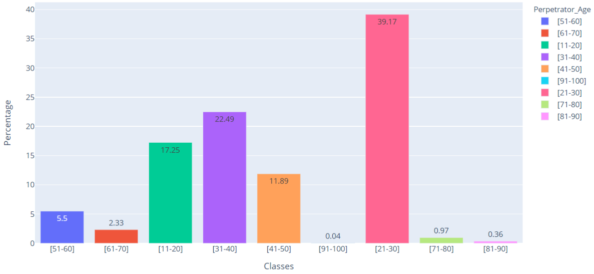


Figure 3.3: Perpetrator Age (9 classes) Distribution

The results obtained, taking into consideration that it was a multi-classification problem with 9 distinct classes, appeared to be excellent and they can be appreciated in Table 3.2. It is important to notice that the RF's parameters had been chosen taking into account the accuracy trend per number of trees (Fig.3.4) and impurity, just as in the previous section. As expected, the best performing algorithm was again the RF, thanks to whom we managed to obtain a classification for all of the 9 distinct labels.

Model	Accuracy	Precision	Recall	F1 score
Decision Tree *(maxDepth = 10, impurity = 'entropy')*	0.2644	**[11-20]**: 0.4952, **[21-30]**: 0.0000, **[31-40]**: 0.0000, **[41-50]**: 0.7769, **[51-60]**: 0.0000, **[61-70]**: 0.0902, **[71-80]**: 0.0000, **[81-90]**: 0.4579, **[91-100]**: 0.4347	**[11-20]**: 0.3413, **[21-30]**: 0.0000, **[31-40]**: 0.0000, **[41-50]**: 0.1809, **[51-60]**: 0.0000, **[61-70]**: 0.2175, **[71-80]**: 0.0000, **[81-90]**: 0.3186, **[91-100]**: 0.6467	0.4041
Random Forest *(NumTrees = 32, impurity = 'entropy')*	0.4211	**[11-20]**: 0.5640, **[21-30]**: 0.1153, **[31-40]**: 0.3377, **[41-50]**: 0.4032, **[51-60]**: 0.2354, **[61-70]**: 0.2315, **[71-80]**: 0.2759, **[81-90]**: 0.6132, **[91-100]**: 0.9894	**[11-20]**: 0.3901, **[21-30]**: 0.3134, **[31-40]**: 0.2413, **[41-50]**: 0.2568, **[51-60]**: 0.3043, **[61-70]**: 0.3665, **[71-80]**: 0.5513, **[81-90]**: 0.7169, **[91-100]**: 0.8878	0.4612

Table 3.2: Perpetrator Age (9 Classes) Classification

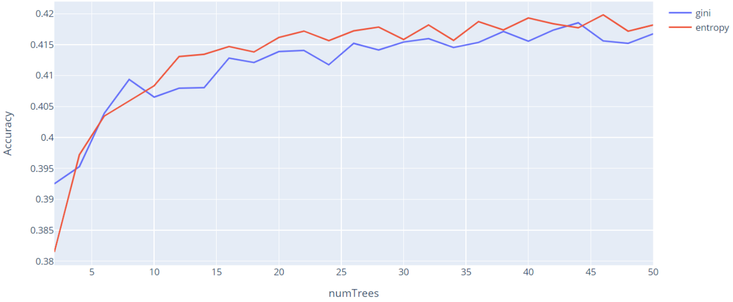
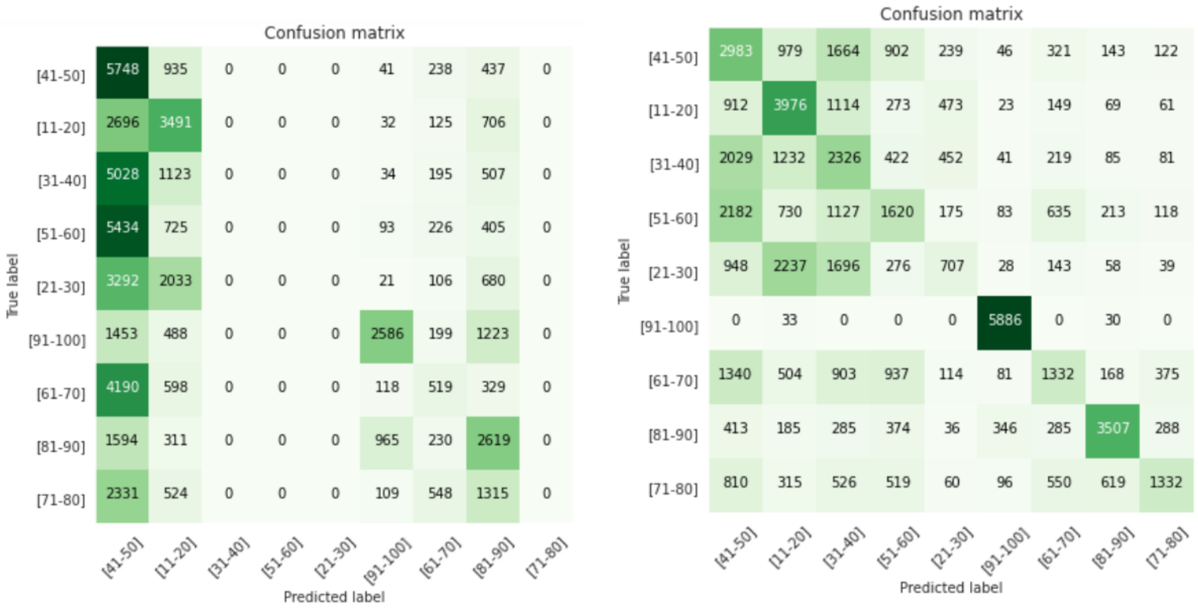


Figure 3.4: RF accuracy per number of tree and impurity

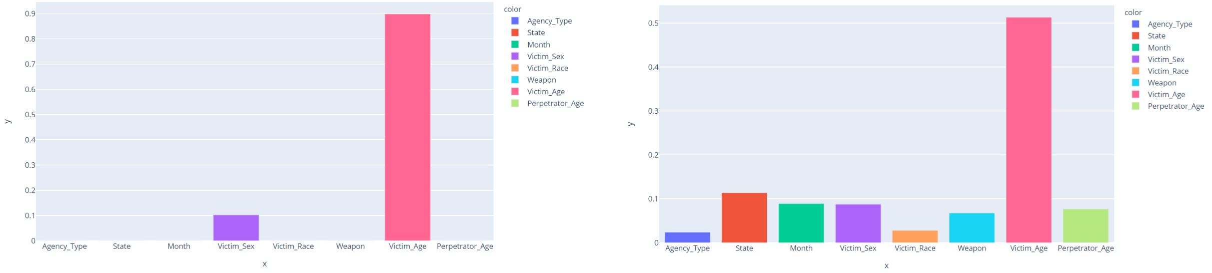
The difference between the results of the RF and the DT could also be detected by observing the two confusion matrices plotted in Figure 3.5: it is in fact highlighted how the RF classifies each class, while the DT fails to consider some of them. This result might be due to the higher complexity of the RF, which manages to take in account all the available features, where instead the DT considers only *Victim Age*, as shown by Figure 3.6. Moreover, we can perceive that in most of the wrongly classified cases the predicted label was the one of the class adjacent to the true one. This can be appreciated especially in the RF classification, since it was able to classify each class. The last interesting remark is that we did not expect some classes to obtain such a high score with both methods, in particular the 'older' ones such as [81-90] and [91-100].



(a) Decision Tree

(b) Random Forest

Figure 3.5: Confusion Matrix per Perpetrator Age (9 classes)



(a) Decision Tree

(b) Random Forest

Figure 3.6: Feature Importance per Perpetrator Age (9 classes)

3.3. Perpetrator Age Classification (4 classes)

In this final task we divided Perpetrator Age in four distinct classes, each one covering a 25 years span, before applying the same two classifiers described in the previous paragraphs, DT and RF. Also in this case, due to the excessively imbalanced distribution between the classes, as shown in Figure 3.7, we decided to apply undersampling and oversampling where needed to restore the balance.

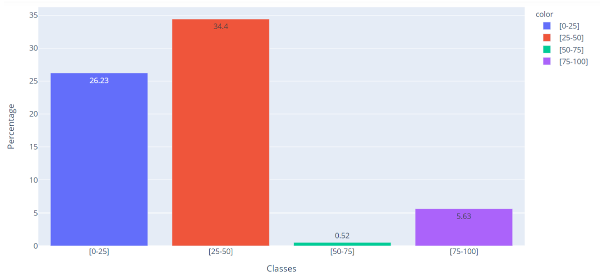


Figure 3.7: Perpetrator Age (4 classes) Distribution

The results obtained are reported in Table 3.3, while the plot of the accuracy trend by number of trees and impurity parameter created in order to verify the model was not overfitting is shown in Figure 3.8. As could be expected, due to the smaller number of target classes all performance measures improved with respect to the ones of the previous subsection. Indeed, here the difference in terms of performance between the DT and RF was less evident than before, since both classifiers successfully classified all the labels. Moreover, also the feature importance were similar and surprisingly the DT used every attribute in its nodes, while *Victim Age* confirmed itself to be the most relevant attribute in the classification.

Model	**Accuracy**	**Precision**	**Recall**	**F1 score**
Decision Tree *(maxDepth = 10, impurity = 'entropy')*	0.5255	**[0 - 25]:** 0.3597, **[25-50]:** 0.6463, **[50-75]:** 0.4093, **[75-100]:** 0.6772	**[0 - 25]:** 0.3999, **[25-50]:** 0.5674, **[50-75]:** 0.4430, **[75-100]:** 0.6686	0.6043
Random Forest *(NumTrees = 28, impurity = 'gini')*	0.5465	**[0-25]:** 0.3978, **[25-50]:** 0.6406, **[50-75]:** 0.4671, **[75-100]:** 0.6743	**[0 - 25]:** 0.4035, **[25-50]:** 0.5811, **[50-75]:** 0.4579, **[75-100]:** 0.7832	0.6094

Table 3.3: Perpetrator Age (4 Classes) Classification

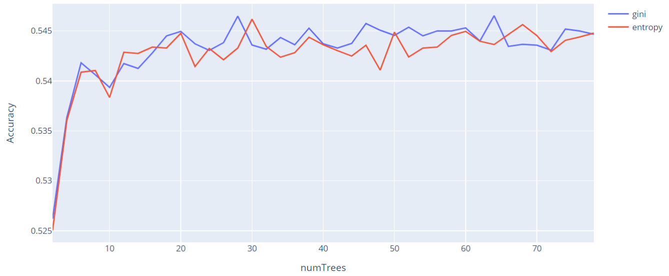
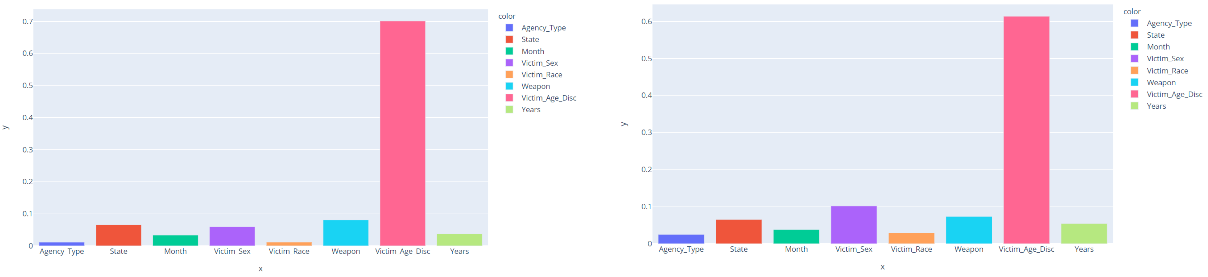


Figure 3.8: RF accuracy per number of trees and impurity

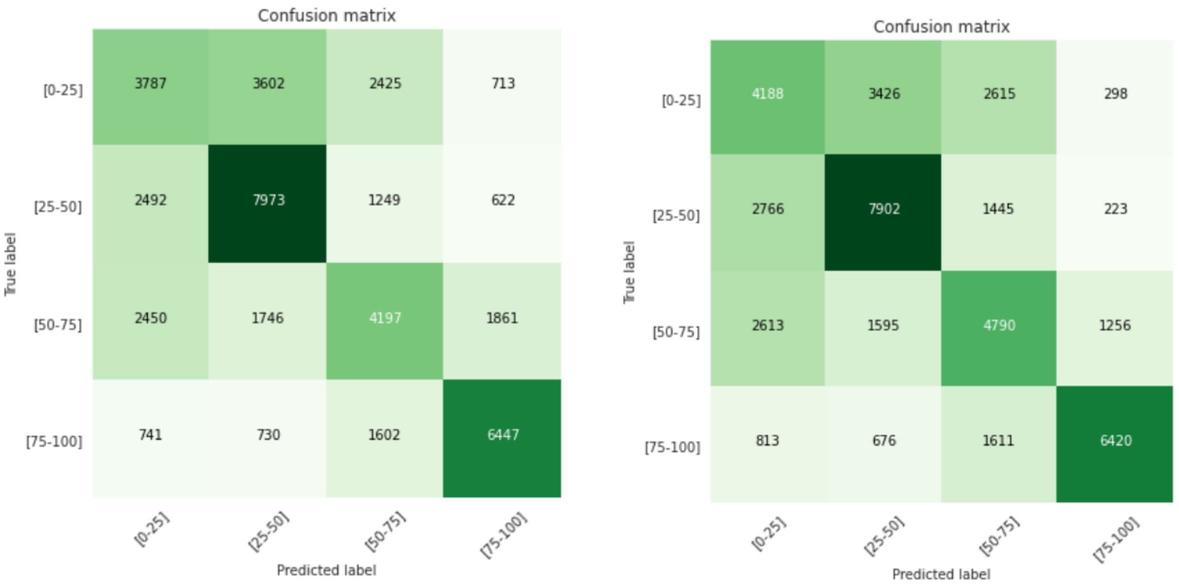


(a) Decision Tree

(b) Random Forest

Figure 3.9: Feature Importance per Perpetrator Age (4 classes)

It could then possible to conclude that, even though the best performances were, as always, achieved by the RF model, this time the DT could be preferred since it is a simpler algorithm but it still lead to excellent results. A confirmation of such statement could be given by the confusion matrices plotted in Figure 3.10, which are now extremely similar for the two classifiers.



(a) Decision Tree

(b) Random Forest

Figure 3.10: Confusion Matrix per Perpetrator Age (4 classes)

A. Python Libraries

In this project we used the following python functions in PySpark.

- SparkContext
- pyspark.sql
  - SQLContext
  - functions
  - SparkSession
  - types
- pyspark.mllib
  - stat (Statistics)
  - classification (LogisticRegressionWithLBFGS)
  - regression (LabeledPoint)
  - evaluation (BinaryClassificationMetrics, MulticlassMetrics)
  - util (MLUtils)
- pyspark.ml
  - feature (IndexToString, StringIndexer, VectorIndexer,Bucketizer, VectorAssembler, OneHotEncoderEstimator)
  - linalg (Vectors)
  - evaluation (BinaryClassificationEvaluator, MulticlassClassificationEvaluator)
  - tuning (CrossValidator, ParamGridBuilder)
  - classification (MultilayerPerceptronClassifier, RandomForestClassifier, DecisionTreeClassifier)
  - Pipeline
- pandas
- plotly
  - graph_objects
  - express
  - subplots
- numpy
- geopy
  - geocoders