# Capstone Project
## Machine Learning Engineer Nanodegree
Arturo Polanco Lozano
September 22, 2016

I. Definition

Project Overview

Image classification is a field of constant development thanks to computer vision and machine learning, it is possible to define the context from a scene in an image improving the performance for object recognition. In this project the goal is to extract features from a database containing images of: Forest, Streets, Coast and Highways; extract the features of these images like color and texture, put them inside a single vector and construct a random forest classifier that can recognize and discriminate amongst these four classes. Random Forest is a machine learning method of randomly constructing and aggregating multiple decision trees to create an even more accurate classifier and due to it's great performace and velocity is the one choosen for this problem.

Problem Statement

Design a scene catigorizator using random forest, the following steps are required in order to do it :

1. Download and preprocess the Antonio Torralba's scene recognition dataset of images.
2. Use HSV's color space to extract the mean and standard deviation to use it as a descriptor.
3. Split the dataset into training and testing data.
4. Initialize Random Forest Classifier and train it.
5. Evaluate the classifier.

The purpose is to use the scene classifier to simplify in future projects related to the task of object recognition.

Metrics

Accuracy is the first metric to be checked when the algorithms are evaluated, is the sum of true positives and the true negative outputs divided by the data size.

$$accuracy = \frac{true\ positives + true\ negatives}{dataset\ size}$$

The scikit-learn library provides a convenience report when working on classification problems to give you a quick idea of the accuracy of a model using a number of measures. The classification report function displays the precision, recall, F1-score and support for each class. Due to the nature of this classification problem I found apropiate to use this report metric in order to have a good idea of how the algorithm is behaving.

Precision (P) is defined as the number of true positives (Tp) over the number of true positives plus the number of false positives (Fp).

$$P = \frac{T_p}{T_p+F_p}$$

Recall (R) is defined as the number of true positives (Tp) over the number of true positives plus the number of false negatives (Fn).

$$R = \frac{T_p}{T_p+F_n}$$

These quantities are also related to the (F_1) score, which is defined as the harmonic mean of precision and recall.

$$F1 = 2\frac{P \times R}{P+R}$$

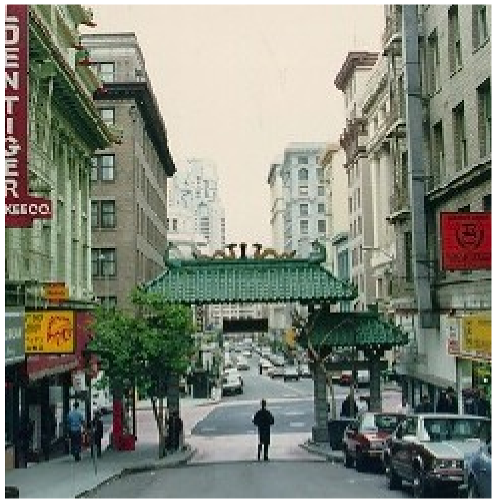
II. Analysis

Data Exploration

Antonio Torralba is a computer vision, machine learning and human visual perception researcher from MIT that provides 4 categories of scenes (Highway, Forest, Coast and Street) in a dataset with 1236 color images using 256 x 256 pixels. The dataset is available for free in the his website and didn't need any kind of cleaning, it does not have any categorical variables, missing values, outliers, etc. . the objects and regions in this dataset have been fully labeled and he annotations are available in LabelMe format.

Exploratory Visualization

The 4 catefories have as main features the constrast of color and the texture, it's possible to take advantage of this since they are different in each of the categories.



Algorithms and Techniques

The classification problems offer a wide variety of machine learning alorithms to solve a problem, it's impossible to know which algorithm is going to work in the best way on the dataset beforehand. That means is possible to discover it using trial and error with a list of algorithms. Some of the must important machine learning algorithms are included in this project and the metric that score it's performance is accuracy: Logistic Regression, K-Nearest Neighbors, Naive Bayes, Random Forest and Support Vector Machines.

Logistic regression (LR) assumes a Gaussian distribution for the numeric input variables and can model binary classification problems. You can construct a logistic regression model using the LogisticRegression class.

K-Nearest Neighbors (KNN) uses a distance metric to find the K most similar instances in the training data for a new instance and takes the mean outcome of the neighbors as the prediction. You can construct a KNN model using the KneighborsClassifier class.

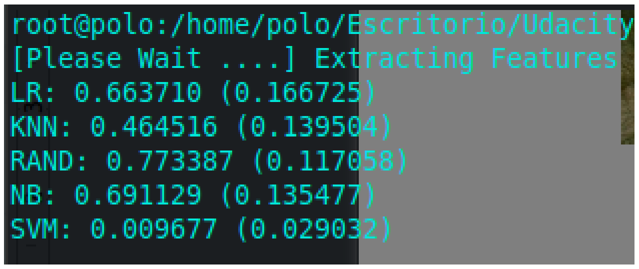
Naive Bayes (NB) calculates the probability of each class and the conditional probability of each class given each input value. These probabilities are estimated for new data and multiplied together, assuming that they are all independent (a simple or naive assumption). When working with real-valued data, a Gaussian distribution is assumed to easily estimate the probabilities for input variables using the Gaussian Probability Density Function. You can construct a Naive Bayes model using the GaussianNB class.

Random Forest (RAND) is an ensemble learning method for classification, regression and other tasks and works constructing many decision trees that outputs the mean prediction of all of them.You can construct a RAND model using the RandomForestClassifier.

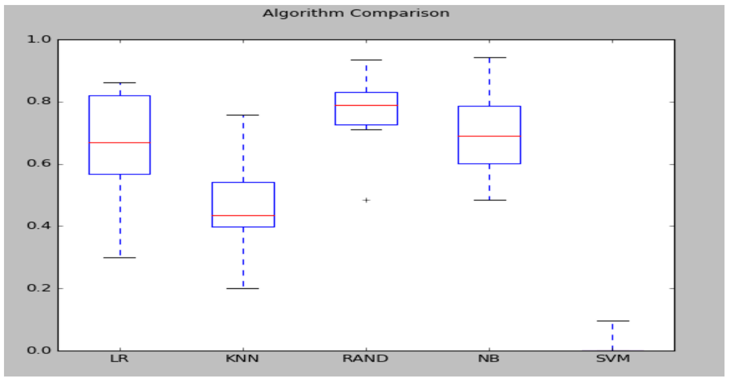
Support Vector Machines (SVM) seek a line that best separates two classes. Those data instances that are closest to the line that best separates the classes are called support vectors and influence where the line is placed. SVM has been extended to support multiple classes. Of particular importance is the use of different kernel functions via the kernel parameter. A powerful Radial Basis Function is used by default. You can construct an SVM model using the SVC class.

Benchmark

Testing the accuracy of the previous algorithms:



```
root@polo:/home/polo/Escritorio/Udacity
[Please Wait ....] Extracting Features
LR: 0.663710 (0.166725)
KNN: 0.464516 (0.139504)
RAND: 0.773387 (0.117058)
NB: 0.691129 (0.135477)
SVM: 0.009677 (0.029032)
```

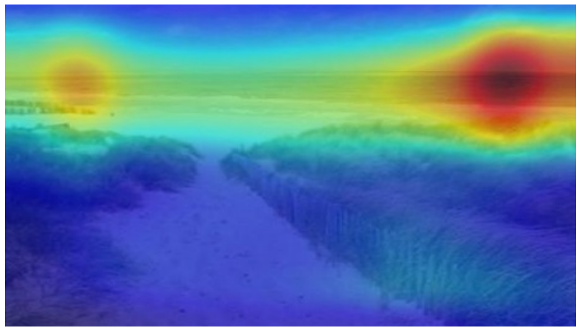


The winner of the test is Random Forest, followed by Naive Bayes and Linear Regression. Thanks to the whisker plots shows how spread the accuracy scores across each cross validation fold for each of them. Comparing the performance of all of them now is clear that with this dataset the best algorithm is Random Fores and is the one

III. Methodology

Data Preprocessing

These four categories of images exhibit a fair amount of variation in color, it is possible to take advantage of it extracting the basic color chanenel statistics from HSV color space, compute the standart deviation for each of the Hue, Saturation and Value components, respectively and concatenate them together; to characterize the texture of the image Haralick Texture Features is applied to form a texture feature vector with 13 entries, after that is used with the color feature vector and the texture feature vector and return the calling function.



Implementation

The first thing that needs to be done is importing all the necessary packages that are needed to solve this problem; then the program creates a function called describe, basically is the descriptor of the image which is going to provide the features of every single image so the classifier can work, this features are the Harlick texture features, the mean and standard deviation in HSV color space of all the channels and returns a concatenated feature vector of color statistics and Harlick texture vectores.

After that once the dataset is requested by the user the image paths is grabbed and then is initialized the list of labels and matrix of features, with a loop over the images the label is extracted and the image is loaded from the disc, then the features are extracted from the image and updates the list of labels and features.

Adtionally the training and testing split is created leaving 75% for training and 25% fro testing, the Random Forest classifier is initialized and the training for the model starts, then the testing outputs the final predictions and compare them with the test labels to measure the Precision, F1-Score and the recall using the classification report class from sklearn metrics.

Finally as a visualization aid some random images are printed with the classification tag given by the alorithm, this task is possible due to open cv, an open source library utilized for computer vision which is very important for projects of object recognition.

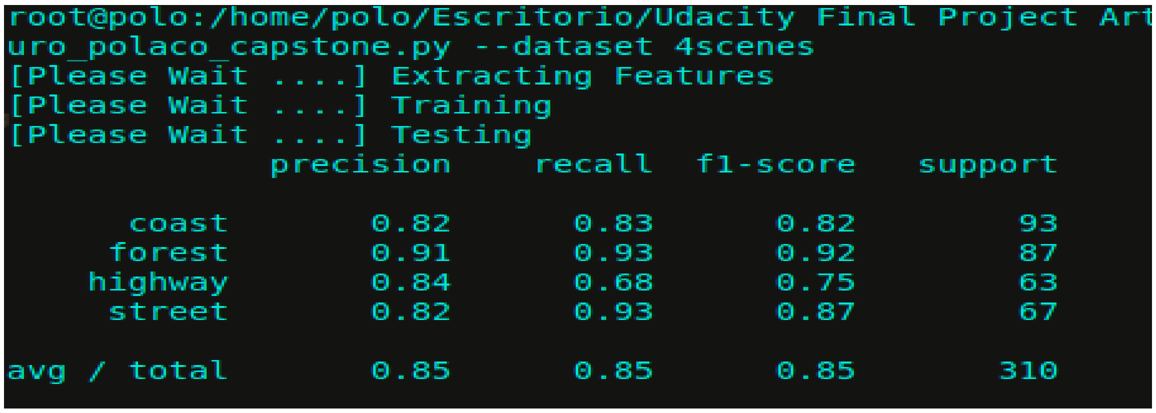
Refinement

The Random Forest Classifier is by itself an improved version of the Decision Trees, usually when we have a good performance using Decision Trees it's worth trying the Random Forest. The parameters that can be used to make improvement are n_estimators that the represent the number of trees used in the forest and random state which is the seed for the random number generator. The improvement was reflected from 0,85 the Accuracy to 0,86.

IV. Results

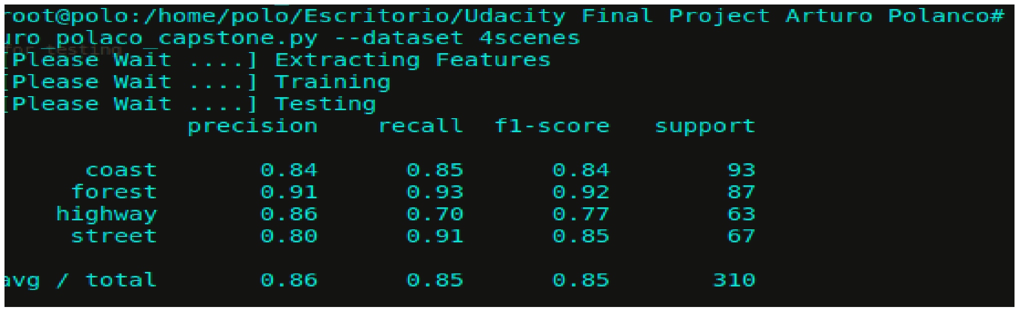
Model Evaluation and Validation

Before of tunning the parameters of the Random Forest classifier:



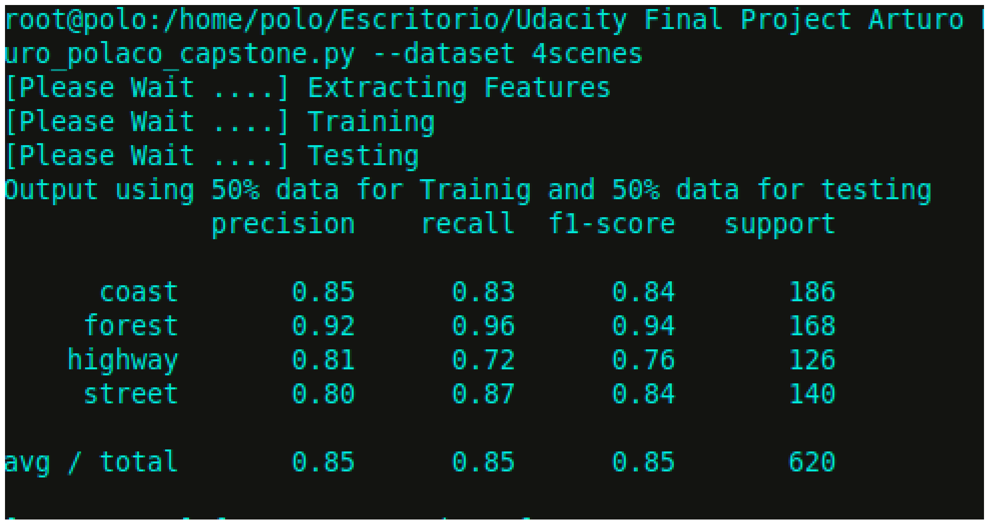
```
root@polo:/home/polo/Escritorio/Udacity Final Project Art
uro_polaco_capstone.py --dataset 4scenes
[Please Wait ....] Extracting Features
[Please Wait ....] Training
[Please Wait ....] Testing
             precision    recall  f1-score   support

      coast       0.82      0.83      0.82        93
     forest       0.91      0.93      0.92        87
    highway       0.84      0.68      0.75        63
     street       0.82      0.93      0.87        67

avg / total       0.85      0.85      0.85       310
```

After tunning the parameter the precision improved a little bit to 0.86, the testing size is 0.25%



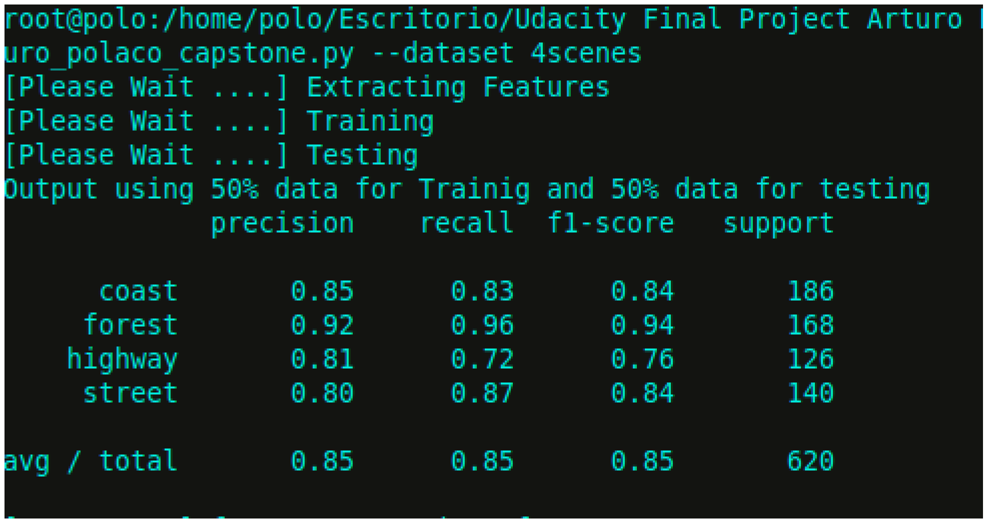
```
root@polo:/home/polo/Escritorio/Udacity Final Project Arturo Polanco#
uro_polaco_capstone.py --dataset 4scenes
[Please Wait ....] Extracting Features
[Please Wait ....] Training
[Please Wait ....] Testing
             precision    recall  f1-score   support

      coast       0.84      0.85      0.84        93
     forest       0.91      0.93      0.92        87
    highway       0.86      0.70      0.77        63
     street       0.80      0.91      0.85        67

avg / total       0.86      0.85      0.85       310
```

The precision is high, the best possible with all the classifiers tested. It's a reasonable value and trust worthy using it for the purpose of improving object recognition tasks. The model meets expectations.



```
root@polo:/home/polo/Escritorio/Udacity Final Project Arturo
uro_polaco_capstone.py --dataset 4scenes
[Please Wait ....] Extracting Features
[Please Wait ....] Training
[Please Wait ....] Testing
Output using 50% data for Trainig and 50% data for testing
             precision    recall  f1-score   support

      coast       0.85      0.83      0.84       186
     forest       0.92      0.96      0.94       168
    highway       0.81      0.72      0.76       126
     street       0.80      0.87      0.84       140

avg / total       0.85      0.85      0.85       620
```

Finally I set the Training and Testing data to 50% the precision decreases again to 0.85, this means it had less data to be trainid and the precision only changed by 0.1%. The model is robust because is not sensitive to changes in the training data..



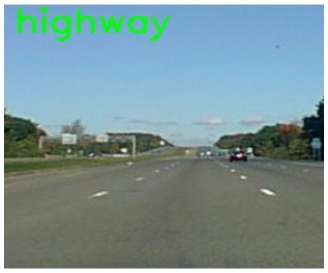
```
root@polo:/home/polo/Escritorio/Udacity Final Project Arturo
uro_polaco_capstone.py --dataset 4scenes
[Please Wait ....] Extracting Features
[Please Wait ....] Training
[Please Wait ....] Testing
Output using 50% data for Trainig and 50% data for testing
             precision    recall  f1-score   support

      coast       0.85      0.83      0.84       186
     forest       0.92      0.96      0.94       168
    highway       0.81      0.72      0.76       126
     street       0.80      0.87      0.84       140

avg / total       0.85      0.85      0.85       620
```

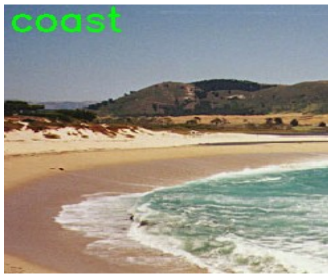
Justification

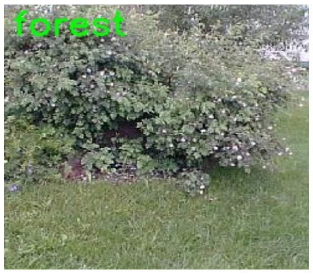
The result of the model was very satisfacoty, comparing to the benchmark the final model either under nor over the expected, is possible to say that the result is trust worthy and the project is finished under satisfaction the precision of 85%. The only thing negative is the time it takes for training, maybe 10 to 15 seconds but the output has correct labels and that is the must important.

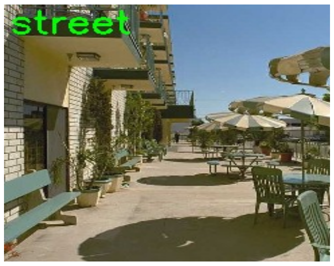
V. Conclusion

Free-Form Visualization









Precision and good performance, this accuracy is attributed to the ensemble method on aggregating weak learners based on decision trees to form a stronger classifier, the random forest. The randomness is introducen into the training data selection and the feature column selection when training each tree in the forest, performing these tho levels of random sampling helps avoid overfitting and generates more accurate classifier.

<u>Reflection</u>
In this project was shown the Random Forest Classifier, an ensemble method that performs usually better than the other ones. Great in classification of computer vision challenges. The project started using as descriptor of the dataset the texture and the variation of colors, a strong feature that varies a lot and could be taken as an advantage to classify them. Then all the features of the training data were extracted and the random forest made the decision of assign each image to it's corresponding category; after that some metrics were applied to evaluate the performance of the model and finally for visualization purposes some pictures were shown with the corresponding label.

<u>Improvement</u>
Finding an appropiate machine learning model is not the end of the work, it's possible to save and load the model using skikit-learn's Pickle, a standard way to serializing objects in python to a file; then is possible to lad the file to deserialized the model and use it to make new predictions. This consideration must be taken in count for improvement of the model.