Capstone project proposal
Naga Maheswara
Machine Learning Engineering Nano Degree March 31, 2018

Galaxy classification

Definition

1.1 Project overview :

Exploring this vast universe through a mere human eye is like trying to pick a drop of water from a ocean, human kind is in an urgent need of finding new machines and methods to explore this field. Classifying the explored objects into identified classes such as planets, stars, galaxies, supernovae etc is a major problem. A more specific problem would be classfiying those objects further i.e classfying stars into red gaints, blue gaints, white dwarfs etc or classifying galaxies into spiral, elliptical, merger etc.

This project is intended to build a model that classifies galaxies as spiral, elliptical and uncertain classes. Generally such classifications are done using images of the galaxies themselves but this project does the classification using expert opinions on those images. Each image of the gaalxy is observed by a group of astronomers and they vote for the shape of galaxy. This information will be used for classification.

We will be using a dataset obtained from galaxyzoo organization at https://data.galaxyzoo.org/. This is a publicly available and free dataset. It contains nearly 0.7 million SDSS[1] galaxies. This is a slightly unbalanced dataset.

[1] ***Sloan Digital Sky Survey** a major multi-filter imaging and spectroscopic redshift survey using a dedicated 2.5-m wide-angle optical telescope at Apache Point Observatory in New Mexico, United States.*

1.2 Problem Statement :

This problem is about bulding a mutliclass classfication model that predicts if the given galaxy is a spiral galaxy or elliptical galaxy or neither of them. All the datapoints have *SPIRAL , ELLIPTICAL and UNKOWN as* result features, which have either 0 or 1 as the possible values. Only one of these three fields contains a ' 1', which indicates that the galaxy falls in that cluster (has that shape). This is a multiclass classification problem as it needs to find the class of the galaxy among three different classes.

Workflow of this project is going to be as follows,

1. The data is obtained from galaxyzoo website at https://data.galaxyzoo.org/ and this dataset is loaded into the model.
2. The data is made suitable for training and testing the model by preprocessing and removing the outliers (if any).
3. The data is divided into appropriate sets for training,validating and testing the model.
4. Model is trained on the dataset with specific algorithms that work on multiclass classification.
5. Trained model is evaluated using the best metrics that suit for the chosen algorithms.

Thus, we end up with a model that classifies the galaxies into spiral, elliptical and uncertain classes.

1.3 Metrics :

As the dataset is unbalanced and the problem is a multiclass classification, f1_score of sklearn appears to be a best choice to evaluate the model as f1_score works well on multiclass classification model and we can't use a general evaluation metric such as accuracy_score over an unbalanced dataset.

F1_score is given by,

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Where precision and recall are given by,

$$P = \frac{T_p}{T_p+F_p} \qquad R = \frac{T_p}{T_p+F_n}$$

Here T_p is true positives, F_p is false positives and F_n is false negatives.

F1_score metric is directly available in sklearn module of python. This is a well defined and standard metric used in such problems.

Analysis

2.1 Data Exploration :

Galaxyzoo dataset has a vast number of galaxies that are calssified based on three criteria namely, 'Clean', 'Super clean', and 'Greater'. When 'Greater' criterion is followed, a galaxy is to be placed in the cluster that corresponds to the feature with highest fraction of votes. When 'Clean' criteria is followed, a galaxy is to be placed in the cluster that corresponds to the feature that has more than 80 % of the votes . When 'Super clean' criteria is followed, a galaxy is to be placed in the cluster that corresponds to the feature that has more than 95% of the votes. The features in the dataset can be described as follows :

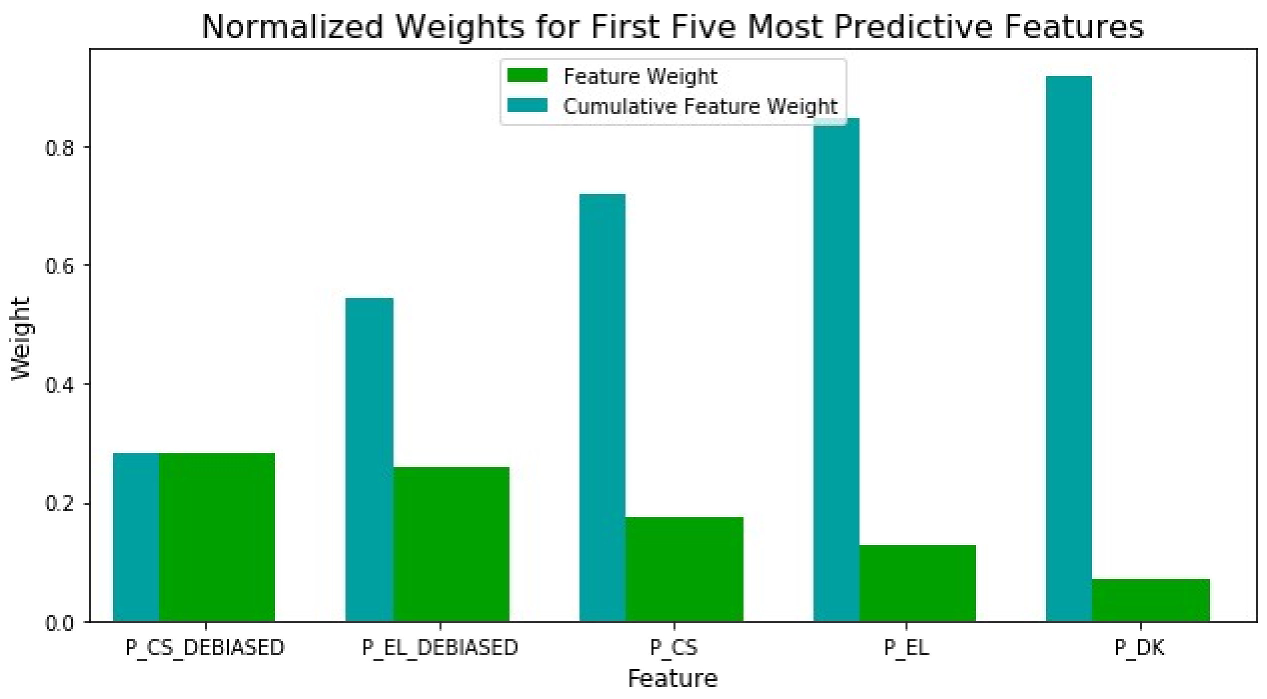
- N_Votes : The number of votes for the object. (This feature doesn't represent any cluster)
- P_EL : The fraction of votes that voted for elliptical.
- P_CW : The fraction of votes for clockwise spiral.
- P_ACW : The fraction of votes for anti-clock wise spiral.
- P_EDGE : The fraction of votes for edge-on spiral.
- P_DK : The fraction of votes for star/don't know.
- P_MG : The fraction of votes for merger.
- P_CS : The fraction of votes for combined spiral (CW + ACW + EDGE).
- P_EL_DEBIASED : The debiased[2] fraction of votes for elliptical.
- P_CS_DEBIASED : The debiased[2] fraction of votes for combined spiral (CW + ACW + EDGE).

This data set is an unbalanced data set it has a large number of uncertain galaxies than spiral or elliptical galaxies. Where more than 50% of the data is uncertain and nearly 35% being spiral and rest being elliptical. This is however not a big problem. But this can be minimized by stratyfying data before splitting. A sample dataset is provided along with this report externally.

[2]Here debiased votes are the corrected votes i.e.. small spiral galaxies appear to be elliptical when their images are faint or if they are distant. Those galaxies are again classified by applying certain corrections and the votes after the corrections are given by debiased votes.

2.2 Explanatory visualisation :

The dataset has other features such as RA, DEC, OBJID etc but such features are not very important for classification. All the features mentioned in the above section are necessary for classification. The following plot shows the first five most important features in training our model.



Above plot shows that P_DK, P_EL, P_CSFEATURE, P_EL_DEBIASED, P_CS_DEBIASED are the most important features for classifying galaxies. This is obviously clear and meets the expectations as these features contain the most important information, as P_EL_DEBIASED, P_CS_DEBIASED, P_DK almost represent the result classes elliptical, spiral and uncertain respectively. These are the features that will be used in building the benchmark model in upcoming sections.

This plot contains features on the horizontal axis, thier weights on the vertical axis. This plot is obtained by training the data on RandomForestClassifier. The weights may change for some other algorithms, however the features stay common in general as they are the most essential features even when building the dataset (as discussed in 2.1 Data Exploration).

2.3 Algorithms and Techniques :

Our approach to this model was trying different algorithms and choose the best algorithm among them. We have very massive datasets in this field. So a large number of new algorithms can be built in this particular context, coming to the current problem, the final output required here is to put the datapoints into different groups which sounds similar to clustering, as our data is pre classified that information will be lost when we consdider this model to be a clustering model. But a data which is labeled to a certain level, adds on the efficiency of a clustering algorithm as it gives a foundation to start with. This is clearly discussed in **"Adapting *k*-means for supervised clustering"** written by S.H.Al-Harbi and V.J.Rayward-Smith and **"Supervised k-Means Clustering"** written by Finley, Thomas, Joachims, Thorsten. Those models mostly rely on parameterizing k-Means and buliding a supervised k-Means using structural SVMs. We do have many such supervised clustering algorithms. But these models can not be implemented in this context as data is huge and completely labeled.

So, the next best alternative available is to choose a supervised classifier. But a normal supervised classifier is not suitable for this problem as the galaxies are to be classified into 3 groups. A normal supervised classifier would work for a binary classification but not for a multi-group classifcation.

The dataset already has the fields *SPIRAL, ELLIPTICAL and UKNOWN*, which clearly show where a galaxy is placed. So, the available dataset is pre-labeled, which implies that a supervised multiclass classifier performs at its best on this dataset. There are several multiclass classification algorithms such as SVM, Decisiontree and Random Forest Classifier, Multilayer Perceptron etc. But the best suited algorithm here, is a RandomForestClassifier because it takes less time to build a Random Forest Classifier than that of building an SVM or MLPClassifier for such large data.The paper "An Empirical Comparison of Supervised Learning Algorithms" by Rich Caruana compared these classifiers and found that Random forest works well over SVM, MLP and Decisiontree Classifiers. Random forest classifier is one of the best supervised multiclass classifiers available. It does a better work on its training and testing times when compared to other classifiers such as SVM or MLP classifiers. This is a crucial aspect to choose an algorithm as this model works on a massive dataset. So, a faster algorithm will always have its advantage here. However MLPClassifier builds a strong network which is more trustworthy and overfitting can be reduced. So we build both the models of RandomForestClassifier and MLPClassifier to check if one does better than the other.

So, we use RandomForestClassifier and MultilayerPerceptron algorithms in this model to train on the dataset. The default parameters passed to classifier models are,

- *max_depth* is the parameter that defines maximum depth to which the random forest can be built.
- *max_features* limits the number of features to be considered while learning.
- *min_samples_leaf* limits the number of samples required to be at a leaf node.
- *max_iterations* which gives the number of iterations that an MLPClassifier can run.
- *Verbose* which decides whether to show the loss through each iteration or not.

These are some of the major parameters used in these algorithms in our model.

RandomforestClassifier can be described as a group of several decision trees. Our model sees every data point builds several decision trees and attains the result by building a random forest.

Multi layer perceptron network is a neural network (deep learning technique) which works excellent in multi class classification. Here we build an architecture for the network and the whole archittecture is initialized with random weights. Now each of the datapoints is made to pass through the architecture, the weights of each link are updated as the datapoint passes through the architecture. Once a huge number of data points pass through the architecture the weights are stabilized in such a way that they can guide a new unlabelled point. Thus attaining our requirement of classifying the galaxies.

2.4 Benchmark :

This Benchmark is set based on the criteria discussed in *Data Exploration* and *Exploratary visualisation*. Here, datapoint will be assigned to a *SPIRAL* cluster if the P_CS_DEBIASED is the highest and is above 50%. The object will be assigned to a *ELLIPTICAL* cluster if the P_EL_DEBIASED is the highest and is above 50%. The object is classified to be *UNKNOWN* if both the features are below 50% (0.5).

This is surely not one of the best criterion but it does classify data to a decent level. This can be used as a benchmark for our model as this is easier and simpler to build and also does a considerable classification.

Methodology

3.1 Data Preprocessing :

Data preprocessing generally involves cleaning the data from abnormalities such as outliers, missing values or unbalanced dataset etc. Galaxies surely have many outliers as we have some of the galaxies which are very huge that they are comarable to 1000 times an average galaxy. However, this is not going to be a great problem in our case as the features we consider do not include size or any image either, rather the dataset consists the votes of astronomers which are free from such outliers. Considering the possibility of outliers in this votes, there is no room for such outliers either, as the votes are wieghted to fall in the range of 0 to 1 and there is a possibility to have a galaxy get all the votes in a single field if it is completely clear that it falls in that field. So, it is common to have the values vary randomly in range of 0 to 1. Thus we do not deal with any outliers here.

We can get rid of missing values (if any) by using fillna method and fill them with mean of the feature. However our dataset is slightly unbalanced as it has more data points under uncertain field. But that is not going to be a great problem as the dataset is not completely unbalanced. It has a fair share of each class. So we need not deal with unbalanced nature of this dataset.

However, we need to construct a new feature which is result feature that contains the data from the three result classes namely *SPIRAL, ELLIPTICAL, UNCERTAIN* so as to build a single result feature that has all the data of three features.

Further, we divide the data into training and testing sets. So as to train the model and evaluate the model in the end, with the respective sets.

3.2 Implementation :

Implementation of our model invloves two major steps namely, training and testinng the model.

- Training :
    - This phase involves dividing the data into appropriate sets as mentioned in the previous section.
    - Then we train the model using two different alogorithms as discussed in earlier section *2.3 Algorithms and Techniques.*
    - We build a randomforestcalssifier using sklearn.ensemble and

train the model on the training dataset.
- Once the model is trained on both the algorithms we move on to testing phase.

- Testing :

  - This phase invovles using test set to evalute the model efficiency.
  - Here we predict the test results through previously trained model and store those results to evaluate.
  - We evaluate this model using f1_score as a metric for the reasons dicussed in the earlier section *1.3 Metrics.*

Repeat the whole process until the required efficiency is attained. Once the model is attains required efficiency, the best algorithm among the both MLPC, RFC is chosen and decided as the final algorithm for the model.

3.3 Refinement :

Initial solution framed to solve this problem is mentioned in *Benchnmark* as considering the datapoint to be in a class for which it got morethan 50% of the votes and uncertain if none of the classes got that percentage of votes.

This Benchmark model attains an efficency around 50%. To improve this efficiency and reach a considerably decent level, we need to worrk on our techinque of classification.

Benchmark classification was clearly not a best technique to be used. So we rely on Random Forest Classifier which is one of the best multiclass classification algorithms. We end up with an efficiency around 85% which is way better than bench mark model. Further we proceed with MLPClassifier as discussed in the workflow. MLPClassifier gives a further improved classification with an efficieny of 89% which is better than that of RandomforestClassifiers.

But these scores can further be improved in both the cases by tuning the parameters in Randomforest Classifier and building more complex yet non overfitting architecture. Where we finally end up with an efficiency which is somewhere between 91 and 93%. This can move further, which is clearly discussed in the coming section *5.3 Improvents.*

Results

4.1 Model Evaluation and Validation :

Current model uses a validation set to validate the model while learning. This model trained on two different algorithms one bieng RandomForestClassifier which is given a max_depth of 17 which is chosen as best after trying different depths and other parameters are assigned default values as changing them do not show any significant changes in the efficiency.

Second algorithm used here is an MLPClassifier. Here we use MLPClassifier from sklearn and It has an architecture which has 9 hidden layers. First hidden with 7 nodes and an last hidden layer with 7 nodes and rest of the 7 hidden layers each having 12 nodes and the central node with 21 nodes. This architecture is attained through several combinations and trails. This runs for around 50 iterations before winding up.

The final result is more comforting as it attains the required evaluation mark. Final model is very stable and robust that it can be fit even over a manipulated dataset. This model becomes so robust as the values of input features are not confined to a fixed pattern and and the dataset is large enough to avoid overfitting on any small patterns found. Thus the model is free from overfitting and is robust, as the dataset is a well distributed dataset. Dataset has been evaluated using f1_score on a test set which has more than a 100,000 unseen data points. Thus the model reacts very well for unseen data.

The efficiency of the model is above 90% and is not overfitted, can acommadate slight variation in the data. So, our model is a trustworthy model.

4.2 Justification :

The final model built using the MLPCLassifier and RandomForestClassifiers does a very good job in classiying the galaxies. The final results are way better than the benchmark model. Final results are 87% and 89% efficient which are very better than benchmark result which is 49% efficient.

Final solution involves two algorithms RandomForestClassifier, MLPCLassifier using sklearn it runs for 50 iterations (approx). Both the models are comparable to each other in efficiency and RandomForestClassifier is much faster than MLPClassifier, but as the dataset is a moderately large yet not extremely large, MLPClassifier also gives the desired results with a considerably less delay.

Final model built is aorund 90% efficient which is a decently good efficiency. So the model is significant enough to classify the galaxies and solve the problem of using higly complex image processing algorithms or heavy machinery to identify details of the galaxy (other than images).This classification can be done with a very less hardware and time utilization when compared to image classification algorithms which are currently used.

Conclusions

5.1 Free-Form Visualization :

This model is completely involved in numerical data, So a solid pictorial or visual representation is not appropriate in this context. However, a visualization about feature importance can be provided which was already given in *2.2 Exploratry visualisation.* The loss through each iteration of MLPClassifier of sklearn can be given as,

```
Iteration 1, loss = 0.46208502
Iteration 2, loss = 0.34458956
Iteration 3, loss = 0.33094074
Iteration 4, loss = 0.32600171
Iteration 5, loss = 0.32440589
Iteration 6, loss = 0.32138242
Iteration 7, loss = 0.32190721
Iteration 8, loss = 0.32078004
Iteration 9, loss = 0.32016812
Iteration 10, loss = 0.32025553
Iteration 11, loss = 0.31946809
Iteration 12, loss = 0.31884973
Iteration 13, loss = 0.31848547
Iteration 14, loss = 0.31828256
Iteration 15, loss = 0.31664640
Iteration 16, loss = 0.31725579
Iteration 17, loss = 0.31581234
Iteration 18, loss = 0.30869621
Iteration 19, loss = 0.29663487
Iteration 20, loss = 0.27347524
Iteration 21, loss = 0.25896709
Iteration 22, loss = 0.25578347
Iteration 23, loss = 0.25426037
Iteration 24, loss = 0.25363075
Iteration 25, loss = 0.25152226
Iteration 26, loss = 0.25050431
Iteration 27, loss = 0.25073712
Iteration 28, loss = 0.24998008
Iteration 29, loss = 0.24905436
Iteration 30, loss = 0.24854899
Iteration 31, loss = 0.24781218
Iteration 32, loss = 0.24816483
Iteration 33, loss = 0.24659880
Iteration 34, loss = 0.24679631
Iteration 35, loss = 0.24622637
Iteration 36, loss = 0.24671784
Iteration 37, loss = 0.24615088
Iteration 38, loss = 0.24550244
Iteration 39, loss = 0.24671166
Iteration 40, loss = 0.24487710
Iteration 41, loss = 0.24483511
Iteration 42, loss = 0.24486055
Iteration 43, loss = 0.24389490
Iteration 44, loss = 0.24469317
Iteration 45, loss = 0.24432824
Iteration 46, loss = 0.24417878
Training loss did not improve more than tol=0.000100 for two
consecutive epochs. Stopping.
```

These visualizations speak about the architecture and complexity of the networks built. However, the architecture of the network is dicussed in previous section *4.1 Model Evaluation and Validation.*

5.2 Reflection :

Actual aim behind the project is to provide a classification model that classifies galaxies more efficiently using less hardware and time than current image processing models. As they require a large amount of memory to store images of galaxies and a lot of hardware to process them and train on them, but this model uses mere human votings, which is completely numerical and consumes very less space and computaional time than the prior models.

Summary of the project can be given as,

- Field of interest where machine learning has a hge application is chosen (Astronomy).
- A potential problem which has a practical application is chosen (Galaxy classification).
- A relevant, ethical dataset is obtained (from https://data.galaxyzoo.org/).

- Dataset is analyzed and naotable observations are noted such as the missing values, outliers, balance among the classes etc.
- Algorithms that suit the dataset and problem statement are analyzed.
- A benchmark model is created for the model to rely on.
- Dataset is preprocessed, model was built to train on preprocessed dataset using selected algorithms (RandomForestClassifier, MLPClassifier).
- The trained model is evaluated for its efficiency over an unseen test dataset.
- Several parameter tunings and architectures were tried to attain the best efficiency possible.
- A satisfiable solution is obtained at last that classifies the galaxies with an efficiency around 90%.

Most interesting step through this whole process was researching and analyizing several algorithms to find a best algorithm that suits the situation. I was exposed to many different and interesting algorithms such as "supervised clustering" which uses parameterized k-means algorithm along with SVM which was pretty interesting. The next interesting aspect was searching for the dataset, there I came across several datasets which consisted millions of stars, glaxies, supernovae etc. I was exposed to very huge data sets that contained tens and hundreds of millions of images.

Most difficult part of this project was finding a best architecture that suits the dataset and current scenario. This was difficult as training the model was a time taking process on my currently available device. So I was forced to change the architecture wait for hours to get the result.

Final solution attained is a very good classifier and this solution fits the expectations for the problem I've chosen. This model can be used in general setting to solve similar problems such as classiying stars, planets etc. This can further extend to other fields such as classifiyng different species of animals, birds etc, or classifying genetical families or even classfying the sub-atomic particles.

5.3 Improvement :

This model has attained a decent efficiency of 90%. However, this can further be enhanced using more powewrful hardware which can accomodate more complex architectures yet not too complex that they face a vanishing gradient problem, pushing the efficiency further to reach around 95 – 97%.

Dataset can also be improved by adding more datapoints and taking the votes from a larger group of astronomers than prior group. New algorithms are not necessarily required for this problem, as these algorithms do a decent classification in a considerably comfortable time when operated on a powerful device.

When this model is considered as a benchmark model, we end with a very little room to improve this further because the first benchmark has an efficiency of 49% and has a lot of room to operate on, so that the model reaches a higher efficiency, but second benchmark has already reached an efficiency of 90% leaving us with a very little space to operate on. However, even this space can be utilized to push the model to further efficiency.

However a generalisation of this model can be done using a more generalised dataset that contains all the astronomical entities such as stars, glaxies, supernovae, planets all together. This can be used to classify several objects at once savig hardware and time.