Machine Learning Engineer Nanodegree

Capstone Proposal

Priyank Shah
May 3rd, 2017.

Proposal

Classification of Mushroom as Poisonous or Edible based on its physical traits.

Domain Background

Classification of Mushrooms as poisonous or edible can be included in **healthcare** domain. Healthcare is currently in much need of Machine Learning techniques. There are lots of mundane things which are done manually and take precious time of healthcare personnel. These things can be automated by using machine learning techniques. For example it can be used to classify if a person is obese, underweight or normal by simply measuring certain features of the person and letting the software do the rest instead of manually applying the formulae and then calculating. Another simple example can be to classify various illness or diseases a person could be suffering from by giving the software relevant data such as blood report, full body checkup report etc. This in turn saves time of the doctor to manually go through each report to find the ailment. Furthermore the model could also accurately identify certain disease symptoms which might even be missed out by the doctor.

Coming to the classification of Mushroom, many of us like to go on camping or love to travel and eat new things. The most common thing which can be found in almost any forest is 'mushroom'. At times if the traveler is stranded they can consume them for their survival. Here arises the problem that is that mushroom edible or not. Also not every person has absolute knowledge on mushroom. So for that simple problem you would not want to consult an expert every time or search on Google and find the same species (which can be quite tedious). And when you are in remote places there is high probability that the mushroom you selected can possibly be poisonous and may lead to fatality. As a machine learning model can work remotely without network as long as classification parameters are available, it is quite convenient to install an application that does this. So next time when you are in a situation where you have to consume mushroom for survival, you can use the app and input its features and it will classify it as poisonous or edible. Thus training a simple machine learning model in this can help immensely in the classification. It can also be used when shopping for mushrooms through which you can be absolutely sure that it is edible. This is basically a classification problem and can be easily solved provided relevant and proper data.

I selected this topic so that I could use the experience in adding more such classifications which might be able to save a life in crucial moments. My vision is to make an application which can correctly classify majority of things which are normally available in remote and arid regions. Some of the additions may include types of stem, roots, berries, plants, fruits, fishes, meat, etc. This can be said to be the starting point in a long queue of classification of if something is edible or not.

Following are some research materials on classification:

1) Comparison of different supervised learning algorithms
   : https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf
2) Machine Learning Algorithms: http://scikit-learn.org
3) For Multi-label Classification: http://scikit.ml

Problem Statement

Classification of Mushroom as edible or poisonous based on its physical traits. Mushroom are most common occurrences in remote and far off places in nature. People who like to travel and go to these places often find themselves stranded. For their survival they have to consume food available there. Mushroom is one such material. But the problem is has many species spread around the world and many of those are poisonous or unidentified. Thus by using machine learning techniques it can be correctly classified as poisonous or edible. Thus this problem is of binary classification. The classification is based on its physical traits such as colour, shape, etc. If the mushroom has certain specific traits it is classified either edible or poisonous. Many machine learning techniques can be used to classify them. The problem can be measured by normally used metrics such as precision, recall, f-beta, etc.

Datasets and Inputs

This dataset is taken from **Kaggle** (https://www.kaggle.com)
The link for the dataset is as follows:
https://www.kaggle.com/uciml/mushroom-classification
This dataset includes descriptions of hypothetical samples corresponding to **23 species** of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The dataset has features which are entirely categorical in nature. The dataset will be used after being transformed by LabelEncoder or OneHotEncoding. The following describes the columns and its categorical values and what they represent.
Attribute Information: (classes: edible=e, poisonous=p)

cap-shape: bell=b, conical=c, convex=x ,flat=f, knobbed=k, sunken=s
cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s

cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
bruises: bruises=t, no=f
odor: almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s
gill-attachment: attached=a, descending=d, free=f, notched=n
gill-spacing: close=c, crowded=w, distant=d
gill-size: broad=b, narrow=n
gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o ,pink=p, purple=u, red=e, white=w, yellow=y
stalk-shape: enlarging=e, tapering=t
stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s
stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
veil-type: partial=p, universal=u
veil-color: brown=n, orange=o, white=w, yellow=y
ring-number: none=n ,one=o, two=t
ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

The data set has 8124 samples.
Number of samples with classification as ‘e’ i.e. edible = 4208.
Number of samples with classification as ‘p’ i.e. poisonous = 3196.

Solution Statement

The problem states that given a certain features of mushrooms it should be identified as either edible or poisonous. Thus it can be said the problem is of classification type. As the problem is of classification it is necessary to choose a model that is efficient in classification. Some of the models that can be used to solve this problem are but not limited to, Logistic Regression, Decision Trees, SVM, Random Forest, Ensemble methods, etc. The model can be tested on certain metrics such as precision (as it is necessary to maximize the recognition of poisonous mushrooms), recall, f-beta, etc.

Benchmark Model

The benchmark model in this case would be to classify every mushroom as poisonous. Even though it is not a valid method but it still gives 50% accuracy in prediction as it is a binary classification. This gives a precision of 0.5, recall of 0.5 , f-1 score of 0.5 all of which is quite less. Ideally the model should interpret the features of mushroom and then classify weather the mushroom is edible or not. The model can be evaluated based on the f-1 score as it is the harmonium of precision and recall. But more importantly precision should be considered as higher the precision the better the classification of one class, which is of poisonous mushrooms.

Evaluation Metrics

Evaluation Metrics that will be used in this problem are but not limited to,

1. **Accuracy** : Number of correct predictions made as a ratio of all predictions made. It can be calculated by taking the ratio of correct predictions to the total number of predictions. The higher the accuracy the higher the reliability of the model in classification. But in skewed datasets it is not recommended .
2. **Precision** : Number of true positives (TP)(i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives(TP) and false positives(FP), which are items incorrectly labeled as belonging to the class). It can be calculated by the following expression **(TP/(TP+FP)).**It is normally used when classification of one class precedes over the other classes. In this model precision will have more importance.
3. **Recall** : Number of true positives(TP) divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives(FN), which are items which were not labeled as belonging to the positive class but should have been).It can be calculated by the following expression **(TP/(TP+FN))**.It is used to measure the sensitivity of the model.
4. **F-1 score** : The F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0.It can be calculated by using the following expression **f1 = 2*(p*r/(p+r))**.

Project Design

Project workflow will be as follows:

1. **Accumulation of Data**: Data will be acquired through Kaggle. This will be done either manually downloading and then uploading in the notebook or using urllib or urlretrieve packages.

2. **Data Analysis** : Data will be analyzed for its consistency , if there are any missing values , if there is skewed classes. The analysis will also include segregated count of each class. Types of categorical values.

3. **Data Transformation** : The analyzed data will then be cleaned out (removing incomplete data , inconsistent data). The data will then be transformed into numerical values by using either LabelBinarizer or OneHotEncoding.

4. **Data Split** : Data will be then split into three parts namely training , validation , testing data. Data split will normally be 75, 15 and 10 % respectively. But it can differ.

5. **Defining Models**: Tentative models will be defined that will be used in the training. The models which will be used in training are LogisticRegression, DecisionTreeClassifier, RandomForest, AdaBoost Ensemble, xgBoost ensemble and SVM.

6. **Training the models on the train set**: Each of the above models will be trained on the training set. Then they will be compared with the metrics on validation set.

7. **Selecting the best model**: The best model will be selected based on the performance of the model on test set. The performance will be primarily based on precision and f1 score with secondary importance on recall and accuracy.

8. **Feature analysis**: Analysis will be then done on the features which mostly govern the classification. This includes PCA and ICA analysis.

9. **Retrain with principle features**: Model will then be retrained with the principle features and then be compared.