Food Predict Type Dataset cataloging metadata for machine learning application report

A project
submitted to the department of Computer Science and Engineering
in partial fulfillment of the requirements
for the degree of

Bachelor of Science in Computer Science and Engineering

By

Md: MERAJUL ISLAM

ID: 15163203017

INDEX

CONTENTS **PAGE NO.**

LIST OF FIGURES

Acknowledgement:

First and foremost, we are grateful to the Allah, the Almighty, the Merciful without whose patronage and blessing this project would not have been successfully completed .He gave us zeal, confidence, power of determination and courage and vanquished all the stumbling hardness that we faced on the way .It is an auspicious occasion for us as students of Department of Computer Science and Engineering, one of the prestigious academic centers of the Bangladesh University of Business and Technology, to express our deep feelings of gratitude to the department and especially to our supervisor,

I would like to express my teacher to Nadia afrin ritu for his supervision, special guidance, suggestions, and encouragement through the development of this work.

Abstract:

As the field of machine learning (ML) matures, two types of data archives are developing: collections of benchmark data sets used to test the performance of new algorithms, and data stores to which machine learning/data mining algorithms are applied to create scientific or commercial applications. At present, the catalogs of these archives are ad hoc and not tailored to machine learning analysis. This paper considers the cataloging metadata required to support these two types of repositories, and discusses the organizational support necessary for archive catalog maintenance.

Introduction:

With the wide usage of computers and internet, there has recently been a huge increase in publicly available data that can be analyzed. Be it online sales information, website traffic, or user habits, data is generated everyday. Such a large amount of data present both a problem and an opportunity. The problem is that it is difficult for humans to analyze such large data. The opportunity is that this type of data is ideal for computers to process, because it is stored digitally in a well-formatted way, and computers can process data much faster than humans. The concept of machine learning is something born out of this environment. Computers can analyze digital data to find patterns and laws in ways that is too complex for a human to do. The basic idea of machine learning is that a computer can automatically learn from experience (Mitchell, 1997). Although machine learning applications vary, its general function is similar throughout its applications. The computer analyzes a large amount of data, and finds patterns and rules hidden in the data. These patterns and rules are mathematical in nature, and they can be easily defined and processed by a computer. The computer can then use those rules to meaningfully characterize new data. The creation of rules from data is an automatic process, and it is something that continuously improves with newly presented data. Applications of machine learning cover a wide range of areas. Search engines use machine learning to better construct relations between search phrases and web pages. By analyzing the content of the websites, search engines can define which words and phrases are the most important in defining a certain web page, and they can use this information to return the most relevant results for a given search phrase (Witten et al., 2016). Image recognition technologies also use machine learning to identify particular objects in an image, such as faces (Alpaydin, 2004). First, the machine learning algorithm analyzes images that contain a certain object. If given enough images to process, the algorithm is able to determine whether an image contains that object or not (Watt et al., 2016). In addition, machine learning can be used to

understand the kind of products a customer might be interested in. By analyzing the past products that a user has bought, the computer can make suggestions about the new products that the customer might want to buy (Witten et al., 2016). All these examples have the same basic principle. The computer processes data and learns to identify this data, and then 2 uses this knowledge to make decisions about future data. The increase in data has made these applications more effective, and thus more common in use

Motivation:

“Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the task or tasks drawn from the same population more efficiently and more effectively the next time."

Machine learning is concerned with the process of constructing abstractions of the real world (concepts, functions, relations and ways of acting) automatically from observations.

Many Algoritham is widely known and used in many businesses to support decision making process and risk analysis. It is also one of legendary learning model which is heavily used in ’60–80’s to build expert systems. One of very popular expert systems which adopt decision tree (almost cs and informatics student knew about) is which developed at 1970 by Buchanan and Cohen. But, like another classic expert systems, Mycin is not fully automatic operated. At that years, human experts still needed to input hard coded rules into expert systems. After 80’s, this model has been lost popularity since it’s seems can not be extended using more sophisticated mathematics.

Problem Description:

Different types of food contain various nutritional elements. Some foods contain more calories than the other. Some are rich in Vitamin A or C. On the contrary, few items contain lots of fat.
In this problem, we have a data-set containing some nutritional facts of different foods. We have 4 classes of food type : Fruit, Vegetable, Meat and Dairy. Our aim is to predict the class of food based on their nutritional elements.

Objective:

The name of the dataset is Food Type Dataset. This dataset contains 8 features which are used to predict the type of the food. The features are nutritional elements of different types of food such as - calories, fat, vitamin, sodium, potassium etc.
For example, apple contains some specific nutritional elements for which it is classified under the class fruit. The other items are similarly classified based on their nutritional features.

Literature review :

Amongst the 5 models that we have used in this project, Random Forest Classifier gives the best result. Random Forest Classifier works with

randomly selected 2 or 3 the features and fit them in the model which results better prediction. There is no guarantee that standardization will improve the classification performance. Standardization will change the distribution of data. Ensemble learning helps improve machine learning results by combining several models.

Dataset Description

The data-set consists of 9 columns among which 8 are features and the last column is the target class. The features are nutritional elements of different types of food. Each class of the food is predicted on the basis of their nutritional facts.

Data Dictionary:

Variable	Definition
Calories	The amount of calorie
Total Fat	Total amount of fat content
Na	Amount of Sodium
K	Amount of Potassium
Protein	Amount of protein
Vitamin A	Percent Daily Value of Vitamin A
Vitamin C	Percent Daily Value of Vitamin C
Iron	Percent Daily Value of Iron

Item	Type of the Food item

Description of the Models

We have used 5 models in this problem to see which one gives the better accuracy. Score for each model is tested individually. The data-set is first standardized by using scalar transform so that no feature has more priority over the other. Then it is split for training and testing. 75% data is used for training and the rest 25% is used for testing purpose.

Sources

1. Fruit Data [2]
2. Vegetable Data [4]
3. Meat Data [3]
4. Dairy Data [1]

Number of Instances

100 (25 in each of the four classes)

Number of Attributes

8 numeric, predictive attributes and the target class is Item.

Attribute Information

1. Calories in cal
2. Total Fat in gm
3. Na (Sodium) in mg

4. K (Potassium) in mg
5. Protein in g
6. Vitamin A in %DV (Percent Daily Values)
7. Vitamin C in %DV (Percent Daily Values)
8. Iron in %DV (Percent Daily Values)
9. Item:
    - Fruit
    - Vegetable
    - Meat
    - Dairy

Missing Attribute Values

No attribute value is missing.

Summary Statistics:

	Calories(g)	Total Fat(g)	Na(mg)	K(mg)	Protein(g)	Vitamin A	Vitamin C	Iron
MAX	300	48.2	850	620	26.2	448	240	15
MIN	10	0	0	19	0	0	0	0
MEAN	118.87	8.191	68.58	229.69	7.779	35.135	25.11	3.173
SD	91.6480	10.7121	103.8913	109.9815	8.5275	64.5198	47.1470	3.6927

Class Distribution

25% for each of 4 classes.

Methodology:

- **Entropy**

  It is a fundamental theorem which commonly used in information theory to measure important of information relative to its *size*. Let x

is our training set contains positive and negative examples, then the entropy of x relative to this classification is:

$$H(x) = - \ p_+ \ log_2 \ p_+ - p_- \ log_2 \ p_-$$

- **Information Gain**

In multivariate calculus, we have learn how to use a partial derivative of each variable relative to all other variables to find local optimum. In information theory, we used similar concept, we derive the original entropy of population to measure information gain of each attribute. For training set x and its attribute y, the formula of Information Gain is:

$$G(x, y) = H(x) - \sum_{i \in value(y)} \frac{|\Delta y_i|}{|\Delta y|} \ H(y_i)$$

I will give you an easy example in order to make sense the formula above. Suppose we face with binary classification ‘yes’ or ‘no’, then we label of bit 1 for yes, and label of bit 0 for no. One of our feature’s attributes is ‘Outlook’ (O) which has three possible values ‘Sunny’ (Os), ‘Overcast’ (Oo), and ‘Rain’ (Or). Then, the information gain of Outlook is:

$$G(x, Outlook) = H(x) - \frac{|Os_1 - Os_0|}{|O_1 - O_0|} H(Os) - \frac{|Oo_1 - Oo_0|}{|O_1 - O_0|} H(Oo) - \frac{|Or_1 - Or_0|}{|O_1 - O_0|} H(Or)$$

Next node is an attribute *Humidity* which has two possible values {*High*, *Normal*}. A branch *High* dominated by single label which is *No*, caused this branch ended with a leaf contains label *No*. Same case with branch *Normal* which ended with a leaf contains label *Yes*.

Decision tree is a very simple model that you can build from starch easily. One of popular Decision Tree algorithm is ID3. Basically, we only need to construct tree data structure and implements two mathematical formula to build complete ID3 algorithm.

SGD (Stochastic Gradient Descent) Classifier

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. SGD has been successfully applied to largescale and sparse machine learning problems often encountered in text classification and natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than 10^5 training examples and more than 10^5 features. The advantages and disadvantage of Stochastic Gradient Descent are as follows:

Advantages:

1. Efficiency.
2. Ease of implementation (lots of opportunities for code tuning).

Disadvantages:

1. SGD requires a number of hyper parameters such as the regularization parameter and the number of iterations.
2. SGD is sensitive to feature scaling.

Decision Tree Classifier

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.
A decision tree has 2 kinds of nodes:

1. Each leaf node has a class label, determined by majority vote of training example searching that leaf.
2. Each internal node is a question on features. It branches out according to the answers.

Random forest Classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees habit of over fitting to their training set.

- Random forest classifier will handle the missing values.
- When we have more trees in the forest, random forest classifier wont over fit the model.
- In the random forest classifier, the higher the number of trees in the forest gives the high accuracy results.

Random Forest Classifier (n_ estimators=200)

n estimators: integer, optional (default = 10)
The number of trees in the forest.

AdaBoost Classifier

Ada-boost, like Random Forest Classifier is another ensemble classifier. Ensemble classifier are made up of multiple classifier algorithms and whose output is combined result of output of those classifier algorithms. Ada-boost classifier combines weak classifier algorithm to form strong classifier. A single algorithm may classify the objects poorly. But if we combine multiple classifiers with selection of training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for overall classifier.

1. Retrains the algorithm iteratively by choosing the training set based on accuracyof previous training.
2. The weight-age of each trained classifier at any iteration depends on the accuracyachieved.

SVM Classifier

Support vector machines (SVMs) are a set of supervised learning methods used for classification and regression. The advantages and disadvantages of support vector machines are given below:

Advantages:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: Different Kernel functions can be specified for the decision function.
  Common kernels are provided, but it is also possible to specify custom kernels.

Disadvantages:

- If the number of features are much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.

- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

C-Support Vector Classification:

The implementation is based on libsvm. The fit time complexity is more than quadratic with the number of samples which makes it hard to scale to dataset with more than a couple of 10000 samples. svm.SVC(kernel=rbf) kernel : string, optional (default =rbf)

Specifies the kernel type to be used in the algorithm. It must be one of linear, poly, rbf, sigmoid, precomputed or a callable. If none is given, rbf will be used.

Implementation:

Csv Food Category Data

Calories(g)	TotalFat(g)	Na(mg)	K(mg)	Protine(g)	VitaminA	VitaminC	Iron	Item
130	0	0	260	1	2	8	2	Fruit
50	4.5	0	140	1	0	4	2	Fruit
110	0	0	450	1	2	15	2	Fruit
50	0	20	240	1	120	80	2	Fruit
60	0	0	160	1	35	100	0	Fruit
90	0	15	240	0	0	2	0	Fruit
50	0	30	210	1	2	45	2	Fruit
90	1	0	450	1	2	240	2	Fruit
15	0	0	75	0	0	40	0	Fruit
20	0	0	75	0	0	35	0	Fruit
60	0.5	0	250	1	8	15	2	Fruit
80	0	0	250	1	2	130	0	Fruit
60	0.5	0	230	1	6	15	2	Fruit
100	0	0	190	1	0	10	0	Fruit
50	0	10	120	1	2	50	2	Fruit
70	0	0	230	1	8	10	2	Fruit
50	0	0	170	1	0	160	2	Fruit
100	0	0	350	1	2	15	2	Fruit
50	0	0	160	1	6	45	0	Fruit
80	0	0	270	1	30	25	4	Fruit
50	1.5	15	180	1	8	110	2	Fruit
40	4	0	250	1	2	130	0	Fruit

50	2.5	20	120	1	2	70	2	Fruit
70	0.3	40	150	0	6	45	0	Fruit
20	0	0	230	2	10	15	2	Vegetable
25	0	40	220	1	4	190	4	Vegetable
45	0.5	80	460	4	6	220	6	Vegetable
25	0	60	250	1	11	10	2	Vegetable
30	0	30	270	2	0	100	2	Vegetable
15	0	115	260	0	10	15	2	Vegetable
10	0	0	140	1	4	10	2	Vegetable
20	0	0	200	1	4	10	2	Vegetable
25	0	20	190	1	0	70	2	Vegetable
10	0	10	70	0	2	8	2	Vegetable
10	0	10	125	1	6	6	2	Vegetable
15	0	35	170	1	130	6	4	Vegetable
20	0	15	300	3	0	2	2	Vegetable
45	0	5	190	1	0	20	4	Vegetable
110	0	0	620	3	0	45	6	Vegetable
10	0	55	190	0	0	30	2	Vegetable
20	0	0	250	1	6	30	2	Vegetable
90	2.5	0	250	4	2	10	2	Vegetable
100	0	70	440	2	120	30	4	Vegetable
25	0	20	340	1	20	40	4	Vegetable
15	2.5	110	225	4	0	8	2	Vegetable

Project Code:

```
import pandas as pd
import numpy as np
from sklearn import metrics


food = pd.read_csv('FoodTypeDataset.csv')

#features from column 1 to 8
feature = food.iloc[:,0:8]
feature = np.array(feature)

#target from column 9
target = food.iloc[:,[8]]
target = np.array(target)

#standarizing the features
from sklearn.preprocessing import StandardScaler
scalerX =StandardScaler().fit(feature)
feature= scalerX.transform(feature)

#function for calculating the performance scores
def getScore(y_test,y_pred,y_train_pred):

    #precision
    prec=metrics.precision_score(y_test, y_pred , average='weighted')

```

```
40
41
42  prec_sum1=rec_sum1=f11_sum=sum_acc_train1=sum_acc_test1=0;
43  prec_sum2=rec_sum2=f12_sum=sum_acc_train2=sum_acc_test2=0;
44  prec_sum3=rec_sum3=f13_sum=sum_acc_train3=sum_acc_test3=0;
45  prec_sum4=rec_sum4=f14_sum=sum_acc_train4=sum_acc_test4=0;
46  prec_sum5=rec_sum5=f15_sum=sum_acc_train5=sum_acc_test5=0;
47
48  #K-fold to split the dataset into 5 folds
49  from sklearn.model_selection import KFold
50  cv=5
51  kf = KFold(n_splits=cv,shuffle=True)
52
53  #splitting the data for training and testing using k fold
54  for train_index, test_index in kf.split(feature,target):
55      X_train, X_test = feature[train_index], feature[test_index]
56      y_train, y_test = target[train_index], target[test_index]
57
58      ##########################################################################################################
59
60      #Model 1: SGDClassifier
61      from sklearn import linear_model
62      clf1 = linear_model.SGDClassifier()
63      clf1.fit(X_train, y_train)
64
65      y_train_pred1 = clf1.predict(X_train)
66      y_pred1 = clf1.predict(X_test)
67
```

```
100     ##########################################################################################################
101     #Model 3: Random Forest
102     from sklearn.ensemble import RandomForestClassifier
103     clf3 = RandomForestClassifier(n_estimators=200)
104     clf3.fit(X_train, y_train)
105
106     y_train_pred3 = clf3.predict(X_train)
107     y_pred3 = clf3.predict(X_test)
108
109     #calculating the precision, recall, f1_score, accuracy score for training and testing
110     prec3,rec3,f13,acc_train3,acc_test3 = getScore(y_test,y_pred3,y_train_pred3)
111
112     #the sum of each precision, recall, f1_score, accuracy score for training and testing from each fold
113     prec_sum3=prec_sum3+prec3
114     rec_sum3=rec_sum3+rec3
115     f13_sum=f13_sum+f13
116     sum_acc_train3=sum_acc_train3+acc_train3
117     sum_acc_test3=sum_acc_test3+acc_test3
118
119
120     ##########################################################################################################
121
122     #Model 4: AdaBoost
123     from sklearn.ensemble import AdaBoostClassifier
124     clf4 = AdaBoostClassifier()
125     clf4.fit(X_train, y_train)
126
```

Comparison of the performance scores:

For testing the performance we have chosen 4 performance matrices which are - precision, recall, F1 and accuracy score. The comparison of the performance scores for each model is shown below:

	Precision Score	Recall Score	F1_Score	Accuracy Score For Training	Accuracy Score For Testing
SGD Classifier	0.8634	0.87	0.8867	0.9175	0.87
DecisionTree Classifier	0.9092	0.87	0.8627	1.0	0.87
RandomForest Classifier	0.9258	0.92	0.9202	1.0	0.92
Adaboost Classifier	0.5636	0.61	0.6685	0.7275	0.61
SVM Classifier	0.8451	0.79	0.7910	0.91	0.79

Future Work:

Another area that future research can improve is the variety of the machine learning methods. This research used linear regression, decision trees, and the naïve Bayes classification. Other methods, such as clustering and artificial neural networks can be used to have a better understanding of the importance of method selection. Final area that can be improved is the process of feature creation. Since the data is limited, the amount of feature modification that can be made is also limited. Both data sources used in this research consists of a single table, and custom variables were created using variables from the same table. With a more comprehensive data set that spans multiple tables, there will be more potential to create new custom variables, while keeping in mind that the more a custom variable is, the more difficult it is to interpret the relation between it and the dependent variable.

References:

https://www.kaggle.com/

https://scholar.google.com/

https://link.springer.com/search?facet-subject=%22Food+Science%22

https://www.nriol.com/health/calorie-chart.asp

https://www.healthline.com/nutrition/zero-calorie-foods#section20