TECHNO INTERNATIONAL, NEWTOWN

(FORMERLY KNOWN AS TECHNO INDIA COLLEGE OF TECHNOLOGY)

NEW TOWN, RAJARHAT, KOLKATA - 700156

Department of Information Technology

Name - Shounak Sengupta

Dept - IT 4th Year

Roll – 10

Subject – Project Report

Submitted on - 25/03/2021

INDEX

What is Machine Learning?

- Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions.

- Machine Learning at its most basic is the practice of using algorithms to parse data, learn from it, and then make a determination or prediction about something in the world.

- Machine learning is the science of getting computers to act without being explicitly programmed.

- Machine learning is based on algorithms that can learn from data without relying on rules-based programming.

- Machine learning algorithms can figure out how to perform important tasks by generalizing from examples.

- The field of Machine Learning seeks to answer the question “How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?”

SUPERVISED LEARNING

- Supervised learning is the machine learning task of inferring a function from labelled training data.
- The training data consist of a set of training examples.
- In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value.
- A supervised learning algorithm analyses the training data and produces an inferred function, which can be used for mapping new examples.
- An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.
- This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

UNSUPERVISED LEARNING

- Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from "unlabelled" data (a classification or categorization is not included in the observations).

- Since the examples are unlabelled, there is no evaluation of the accuracy of the structure that is output by the relevant algorithm, which is one way of distinguishing unsupervised learning from supervised learning and reinforcement learning.

- A central case of unsupervised learning is the problem of density estimation in statistics, though unsupervised learning encompasses many other problems (and solutions) involving summarizing and explaining key features of the data.

Machine Learning Algorithms

1. Supervised Learning

How it works: This algorithm consists of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data. Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.

2. Unsupervised Learning

How it works: In this algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means.

3. Reinforcement Learning:

How it works: Using this algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process.

Common Machine Learning Algorithms

1. Linear Regression

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a *X + b.

2. Logistic Regression

It is a classification not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as logit regression. Since, it predicts the probability, its output values lies between 0 and 1 .

3. Decision Tree

It is a type of supervised learning algorithm that is mostly used for classification problems. Surprisingly, it works for both categorical and continuous dependent variables. In this algorithm, we split the population into two or more homogeneous sets. This is done based on most significant attributes/ independent variables to make as distinct groups as possible.

4. SVM (Support Vector Machine)

It is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

For example, if we only had two features like Height and Hair length of an individual, we'd first plot these two variables in two dimensional space where each point has two co-ordinates (these co-ordinates are known as Support Vectors)

5. Naive Bayes

It is a classification technique based on Bayes' theorem with an assumption of independence between predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier would consider all of these properties to independently contribute to the probability that this fruit is an apple.

6. kNN (k- Nearest Neighbors)

It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function.

7. K-Means

It is a type of unsupervised algorithm which solves the clustering problem. Its procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters). Data points inside a cluster are homogeneous and heterogeneous to peer groups.

8. Random Forest

Random Forest is a trademark term for an ensemble of decision trees. In Random Forest, we've collection of decision trees (so known as "Forest"). To classify a new object based on attributes, each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

Packages & Libraries Used

MATPLOTLIB

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter,wxPython, Qt, or GTK+.

There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged .SciPy makes use of matplotlib.

PANDAS

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

It is free software released under the three-clause BSD license. "Panel data", an econometrics term for multidimensional, structured data sets.

Packages & Libraries Used

NUMPY

NumPy is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

The ancestor of NumPy, Numeric, was originally created by Jim Hugunin. NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents.

SCIKIT-LEARN

Scikit-learn is a free software machine learning library for the Python programming language.

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

ROC CURVE

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.
This curve plots two parameters: True Positive Rate. False Positive Rate.

Intermediate Steps for Model Prediction

Data Collection:

We have collected data sets of weather from online website. We have downloaded the .csv files in which information was present.

Data Formatting:

The collected data is formatted into suitable data sets. We check the collinearity with mean temperature. The data sets which have collinearity nearer to 1.0 has been selected.

Model Selection:

We have selected different models to minimize the error of the predicted value. The different models used are Linear Regression Linear Model, Ridge Linear model, Lasso Linear Model and Bayesian Ridge Linear Model.

Training:

The data sets was divided such that x_train is used to train the model with corresponding x_test values and some y_train kept reserved for testing.

Testing:

The model was tested with y_train and stored in y_predict . Both y_train and y_predict was compared.

Shoe Selling Prediction Using Naïve Bayes Classifier

- Naive Bayes algorithm is a supervised learning algorithm, which is used for solving classification problems.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- It is a linear classifier using Bayes Theorem and strong independence condition among features.

Naïve Bayes Classifier

- A data set with n features represented by features: F1,F2,F3,…..,Fn
- Naïve Bayes states, the probability of output Y from features Fi is :

$$P(Y|F1,F2,\ldots,Fn) = P(Y|F1)P(Y|F2)\ldots P(Y|Fn) = \prod_{i=1}^{n} P(Y|Fi)$$

- The requires that the features Fi are conditionally independent. From Bayes Theorem

$$P(Y|Fi) = \frac{P(Fi|Y)P(Y)}{P(Fi)}$$

Naïve Bayes Model for Shoe Selling Prediction

- In this model, we predict whether a shoe will be sold or not based on some of its features.
- The dependent and independent variables are:

  Dependent variable: 'sold'

  Independent variable: 'start price', 'sale price', 'size', 'heel' (low, high, flat, medium).

```
In [5]: dataset.head()
```

Out[5]:

	sold	startprice	saleprice	size	heel
0	0	199.00	NaN	9.5	Low
1	0	375.00	NaN	7.5	High
2	1	299.99	780.0	8.5	High
3	1	49.99	561.0	8.5	High
4	0	89.00	NaN	5.5	Flat

Fig 1. Shoe Selling Dataset

Pre-Processing:

- One Hot Encoding done on 'Heel' column.
- Removing the outliers which are not in between Inter - Quartile Range.
- Filling up the missing values using mean.
- Standardization performed as shoe price and size are in different units.
- Finally predict the output using naïve bayes classifier and calculate the accuracy score.

```
In [9]: preprocessed_dataset['saleprice'].fillna(np.mean(preprocessed_dataset['saleprice']), inplace=True)
        preprocessed_dataset['size'].fillna(np.mean(preprocessed_dataset['size']), inplace=True)

        print(preprocessed_dataset)
```

	startprice	saleprice	size	heel_Flat	heel_High	heel_Low	heel_Medium
0	199.00	330.65312	9.5	0	0	1	0
1	375.00	330.65312	7.5	0	1	0	0
2	299.99	780.00000	8.5	0	1	0	0
3	49.99	561.00000	8.5	0	1	0	0
4	89.00	330.65312	5.5	1	0	0	0
...	...	...	...	...	...	...	...
1095	795.00	720.00000	6.5	0	1	0	0
1096	209.99	330.65312	10.0	0	0	0	0
1097	599.99	599.99000	9.0	0	1	0	0
1098	599.00	330.65312	8.5	0	1	0	0
1099	1050.00	330.65312	4.0	0	0	0	0

[1100 rows x 7 columns]

Fig 2. One Hot Encoding and Filling up Missing values

```
In [11]: import seaborn as sns
         sns.boxplot(x=preprocessed_dataset['startprice'])
Out[11]: <AxesSubplot:xlabel='startprice'>
```

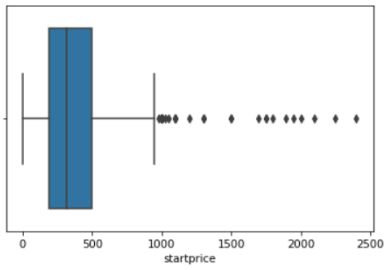

Fig 3. Boxplot to check outliers

```
In [33]: from sklearn.metrics import confusion_matrix, accuracy_score
         print(confusion_matrix(y_test.ravel(), predictions))

         # TP FP
         # FN TN
```

```
[[117   7]
 [ 20  73]]
```

```
In [34]: accuracy_score(y_test, predictions)
Out[34]: 0.8755760368663594
```

```
In [35]: # startprice,saleprice,size,flat,high,low,medium

         print(classifier1.predict(sc.transform([[1500,700,70,0,0,0,1]])))
```

```
[0]
```

Fig 4. Confusion matrix, Accuracy Score and prediction

```
In [37]: from sklearn.metrics import roc_curve
         from sklearn.metrics import roc_auc_score
         from matplotlib import pyplot

         # calculate roc curve
         fpr, tpr, thresholds = roc_curve(y_test.ravel(), predictions)

         pyplot.plot([0, 1], [0, 1], linestyle='--')

         pyplot.plot(fpr, tpr, marker='.')

         pyplot.xlabel('False Positive Rate')
         pyplot.ylabel('True Positive Rate')

         pyplot.show()
```

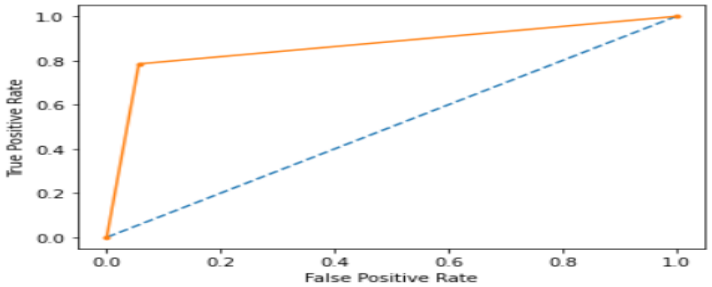

Fig 5. ROC Curve