2581-4575

International Journal For Recent Developments in Science & Technology

Crossref

A Peer Reviewed Research Journal

AGRICULTURAL MARKET PRICE PREDICTION USING MACHINE LEARNING

P.PHANISRI[*1] , S.RAVI TEJA[*2], V.NAGA GEETHIKA[#3] ,D.SAI KRISHNA[#4],

MRS. M. DIVYA SUMITHRA[#5]

[#]1-4 B.Tech, Final year, Department of IT, Gudlavalleru Engineering College, India,

[#]5 Assistant Professor, Department of IT, Gudlavalleru Engineering College, India

ABSTRACT:

What if you could predict whether your stock of choice would rise or fall during the next month? Or if your favorite football team would win or lose their next match? How can you make such predictions? Perhaps machine learning can provide part of the answer. Cortana, the new digital personal assistant powered by Bing that comes with Windows Phone 8.1 accurately predicted 15 out of 16 matches in the 2014 FIFA World Cup. With the help of AZURE we will explore Azure Machine Learning features and capabilities through solving one of the problems that we face in our everyday lives. From the machine learning point of view, problems can be divided into two groups - those that can be solved using standard methods and those that cannot be solved using standard methods. Unfortunately, most real life problems belong to the second group. This is where machine learning comes into play. The basic idea is to use machines to find meaningful patterns in historical data and use it to solve the problem. Crop prices are probably one of the items already in most people's budget. Always the demand and supply should be equal. If demand is greater than supply then there is a shortage otherwise excess. Increase of price leads to lower demand. Inorder to balance the yield and demand, price plays an important role. Constant increase or decrease can influence prices of other groceries and services as well. There are a lot of factors that can influence crop prices, from weather conditions to political decisions and administrative fees, and to totally unpredictable factors such as natural disasters or wars.The actual value and the critical values can be computed by using regression. The predictions are also been compared with the help of residual plots. The residual graphs have been correlated with the predictions.

Keywords: Logistic, Regression,Score,Prediction

Introduction

Data mining is a process of discovering meaningful useful information in large data repositories. Data mining can discover valuable but hidden knowledge from databases. The applications of data mining can be found in many areas such as evaluating risks of financial investment, detection of credit card fraud, patient diagnosis etc.,Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify the content ranges low, medium, and high. A classification task begins with a data set in which the class assignments are known.

In our present investigation, a model that classifies the range levels as low, medium, or high value would also predict the probability of occurrence of existential classifiers of each classification for each datasets of given attributes.

Testing a Model:

A classification model is tested by applying it to test data with known target values and comparing the predicted values with the known values. The test data must be compatible with the data used to build the model and must be prepared in the same way that the build data was prepared. Typically the build data and test data come from the same historical data set. A percentage of the records is used to build the model; the remaining records are used to test the model. Test metrics are used to assess how accurately the model predicts the known values. If the model performs well and meets the business requirements, it can then be applied to new data to predict the future.Regression is a data mining function that predicts a number. The analysis used to model the relationship between one or more independent or predictor values and dependent or response variable .In the present context of data mining the predictor variables or attributes of interest describing the tuple which are known values. The response variable is what we want to predict. A regression task begins with a data set in which the target values are known. In the present investigation the data set values of 'percentage' of nutrient values treated in each group and in each mode of treatment. The two basic types of regression are linear regression and multiple regression. Linear regression uses one independent variable to explain and/or predict the outcome of Y, while multiple regression uses two or more independent variables to predict the outcome.Linear regression assumes a linear or straight line relationship between the input variables (X) and the single output variable (y).More specifically, that output (y) can be calculated from a linear combination of the input variables (X).When there is a single input variable, the method is referred to as a simple linear regression.In simple linear regression we can use statistics on the training data to estimate the coefficients required by the model to make predictions on new data.The line for a simple linear regression model can be written as:

$$y = b0 + b1 * x$$

where b0 and b1 are the coefficients we must estimate from the training data. Once the coefficients are known, we can use this equation to estimate output values for y given new input examples of x.It requires that you calculate statistical properties from the data such as mean, variance and covariance. All the algebra has been taken care of and we are left with some arithmetic to implement to estimate the simple linear regression coefficients.Toestimate statistical quantities from training data and estimate linear regression coefficients from data whichmake predictions using linear regression for new data.A residual plots is scatter plot where the x-axis is the predicted value of x, and the y-axis is the residual for x. The residual is the difference between the actual value and the predicted value of x.

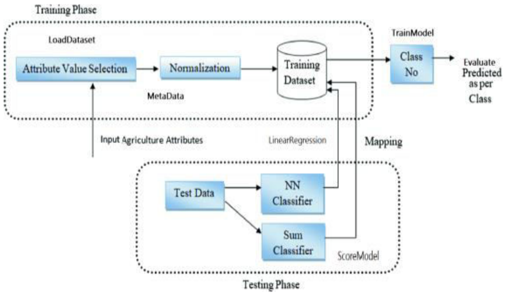
Existing System

The Existing Algorithms provides most important benefits are interpretability. Many affecting diseases infect the Indian rice crop: some diseases are considered more important than others. Moreover the c4.5 can effectively create comprehensive tree with greater predictive power and able to get a prediction error about 1.5% on data of test set. The enhancement in classification results over fitting error using pruning techniques and Handling the huge numbers of attribute values. The Critical values plays important role for pruning the datasets and it may have various classes. The range of sufficiency's are the values derived from the mean ± 4/3 SD and mean ± 8/3 SD (Standard deviation), respectively. The value of lower range <(mean – 8/3 SD) are considered deficient, whereas their low range included all values between > (avg - 8/3 SD) and < (avg – 4/3 SD). Values between > (avg– 4/3 SD) and < (avg + 4/3 SD) are taken as sufficient, whereas the range between > (mean + 4/3 SD) and < (mean + 8/3 SD) are expressed as high. The Higher values concentrations > (mean + 8/3 SD) are expressed as excessive or low.

Proposed system

The plan is to investigate some accessible data and find correlations that can be exploited to create a prediction model.

1. Load Dataset
2. Select columns in Dataset
3.EditMetadata
4. Normalise Data
5. Split Data
6. Linear Regression →Train Model →Score Model
7. Neural Network Regression→Train Model →Score Model
8. Evaluate



I)LoadDataset

Obtaining The Data

Gathering data is one of the most important step in this process. Relevance and clarity of the data are the basis for creating good prediction models. Azure Machine Learning Studio provides a number of sample data sets. Another great collection of datasets can be found at archive.ics.uci.edu/ml/datasets.html.After collecting the data, we need to upload it to the Studio through their simple data upload mechanism. Once uploaded, we can preview the data. The following picture shows part of our data that we just uploaded. Our goal here is to predict the price under the attribute score label.Our next step is to create a new experiment by dragging and dropping modules from the panel on the left into the working area

II)Select Columns in Dataset

This module has no parameters. You use the column selector to choose the columns to include or exclude.

Choose columns by name: There are multiple options in the module for choosing columns by name:

- Filter and search
- Use names in combination with other rules
- Type or paste a comma-separated list of column names

III)MetaData

Typical metadata changes might include:

- Treating Boolean or numeric columns as categorical values
- Indicating which column contains the *class* label, or the values you want to categorize or predict
- Marking columns as features
- Changing date/time values to a numeric value, or vice versa
- Renaming columns

Normalization avoids these problems by creating new values that maintain the general distribution and ratios in the source data, while keeping values within a scale applied across all numeric columns used in the model. This module offers several options for transforming numeric data:

IV)Normalize Data

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information. Normalization is also required for some algorithms to model the data correctly. For example, assume your input dataset contains one column with values ranging from 0 to 1, and another column with values ranging from 10,000 to 100,000. The great difference in the scale of the numbers could cause problems when you attempt to combine the values as features during modeling.

- You can change all values to a 0-1 scale, or transform the values by representing them as percentile ranks rather than absolute values.
- You can apply normalization to a single column, or to multiple columns in the same dataset.
- If you need to repeat the experiment, or apply the same normalization steps to other data, you can save the steps as a normalization transform, and apply it to other datasets that have the same schema.

○ **Zscore**: Converts all values to a z-score.

The values in the column are transformed using the following formula:

$$z = \frac{\mathrm{x} - mean(x)}{stdev(x)}$$

Mean and standard deviation are computed for each column separately. Population standard deviation is used.

○ **MinMax**: The min-max normalizer linearly rescales every feature to the [0,1] interval.

Rescaling to the [0,1] interval is done by shifting the values of each feature so that the minimal value is 0, and then dividing by the new maximal value (which is the difference between the original maximal and minimal values).

The values in the column are transformed using the following formula:

$$z = \frac{\mathrm{x} - min(x)}{[\max (x) \quad - min\ (x)]}$$

○ **Logistic**: The values in the column are transformed using the following formula:

$$z = \frac{1}{1 + \exp(-x)}$$

LogNormal: This option converts all values to a lognormal scale.

The values in the column are transformed using the following formula:

$$z \ = Lognormal.CDF(x;;\mu,\sigma)$$

Here μ and σ are the parameters of the distribution, computed empirically from the data as maximum likelihood estimates, for each column separately.

TanH: All values are converted to a hyperbolic tangent.The values in the column are transformed using the following formula:

$$p(k|x;\theta) = \frac{[E(Y|x)]^k e^{-E(Y|x)}}{k!}$$

V)Split Data

1. Add the Split Data module to your experiment in Studio (classic), and connect the dataset you want to split.
2. For Splitting mode, choose Split rows.
3. Fraction of rows in the first output dataset. Use this option to determine how many rows go into the first (left-hand) output. All other rows will go to the second (right-hand) output. The ratio represents the percentage of rows sent to the first output dataset, so you must type a decimal number between 0 and 1.For example, if you type 0.75 as the value, the dataset would be split by using a 75:25 ratio, with 75% of the rows sent to the first output dataset, and 25% sent to the second output dataset.
4. Select the **Randomized split** option if you want to randomize selection of data into the two groups. This is the preferred option when creating training and test datasets.
5. **Random Seed**: Type a non-negative integer value to initialize the pseudorandom sequence of instances to be used.
6. **Stratified split**: Set this option to **True** to ensure that the two output datasets contain a representative sample of the values in the *strata column* or *stratification key column*

VI) Linear regression

Linear regression is a common statistical method, which has been adopted in machine learning and enhanced with many new methods for fitting the line and measuring error. In the most basic sense, regression refers to prediction of a numeric target. Linear regression is still a good choice when you want a very simple model for a basic predictive task. Linear regression also tends to work well on high-dimensional, sparse data sets lacking complexity. Azure Machine Learning Studio (classic) supports a variety of regression models, in addition to linear regression. However, the term "regression" can be interpreted loosely, and some types of regression provided in other tools are not supported in Studio (classic).

- The classic regression problem involves a single independent variable and a dependent variable. This is called *simple regression*. This module supports simple regression.
- *Multiple linear regression* involves two or more independent variables that contribute to a single dependent variable. Problems in which multiple inputs are used to predict a single numeric outcome are also called *multivariate linear regression*.

The **Linear Regression** module can solve these problems, as can most of the other regression modules in Studio (classic).

- *Multi-label regression* is the task of predicting multiple dependent variables within a single model. For example, in multi-label logistic regression, a sample can be assigned to multiple different labels. (This is different from the task of predicting multiple levels within a single class variable.)This type of regression is not supported in Azure Machine Learning. To predict multiple variables, create a separate learner for each output that you wish to predict.

For years statisticians have been developing increasingly advanced methods for regression. This is true even for linear regression. This module supports two methods to measure error and fit the regression line: ordinary least squares method, and gradient descent.

- **Gradient descent** is a method that minimizes the amount of error at each step of the model training process. There are many variations on gradient descent and its optimization for various learning problems has been extensively studied. If you choose this option for **Solution method**, you can set a variety of parameters to control the step size, learning rate, and so forth. This option also supports use of an integrated parameter sweep.
- **Ordinary least squares** is one of the most commonly used techniques in linear regression. For example, least squares is the method that is used in the Analysis Toolpak for Microsoft Excel.Ordinary least squares refers to the loss function, which computes

error as the sum of the square of distance from the actual value to the predicted line, and fits the model by minimizing the squared error. This method assumes a strong linear relationship between the inputs and the dependent variable.

Score Model for the Linear Regression :

Scoring is also called prediction, and is the process of generating values based on a trained machine learning model, given some new input data. The values or scores that are created can represent predictions of future values, but they might also represent a likely category or outcome. The meaning of the score depends on the type of data you provide, and the type of model that you created. Scoring is widely used in machine learning to mean the process of generating new values, given a model and some new input. The generic term "score" is used, rather than "prediction," because the scoring process can generate so many different types of values:

- A list of recommended items and a similarity score.
- Numeric values, for time series models and regression models.
- A probability value, indicating the likelihood that a new input belongs to some existing category.
- The name of a category or cluster to which a new item is most similar.
- A predicted class or outcome, for classification models.

Train Model for Linear Regression:

Supervised and unsupervised trainingYou might have heard the terms *supervised* or *unsupervised* learning. Training a classification or regression model with Train Model is a classic example of *supervised machine learning*. That means you must provide a dataset that contains historical data from which to learn patterns. The data should contain both the outcome (label) you are trying to predict, and related factors (variables). The machine learning model needs the outcomes to determine the features that best predict the outcomes.During the training process, the data are sorted by outcomes and the algorithm extracts statistical patterns to build the model.*Unsupervised learning* indicates either that the outcome is unknown, or you choose not to use known labels. For example, clustering algorithms usually employ unsupervised learning methods, but can use labels if available. Another example is topic modeling using LDA. You cannot use Train Model with these algorithms.

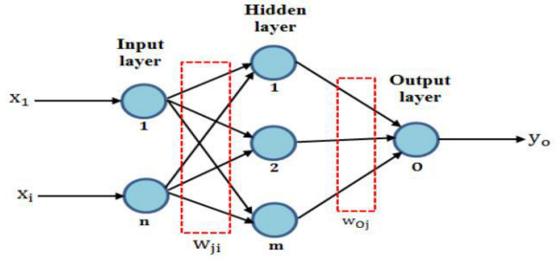
How to use Train Model

Add the Train Model module to the experiment. You can find this module under the Machine Learning category. Expand Train, and then drag the Train Model module into your experiment.On the left input, attach the untrained mode. Attach the training dataset to the right-hand input of Train Model.The training dataset must contain a label column. Any rows without labels are ignored.For Label column, click Launch column selector, and choose a single column that contains outcomes the model can use for training.For classification problems, the label column must contain either categorical values or discrete values. Some examples might be a yes/no rating, a disease classification code or name, or an income group. If you pick a noncategorical column, the module will return an error during training.For regression problems, the label column must contain numeric data that represents the response variable. Ideally the numeric data represents a continuous scale.If you do not specify which label column to use, Azure Machine Learning will try to infer which is the appropriate label column, by using the metadata of the dataset. If it picks the wrong column, use the column selector to correct it.

VII) Create a neural network model

Add the Neural Network Regression module to your experiment in Studio (classic). You can find this module under Machine Learning, Initialize, in the Regression category.

Indicate how you want the model to be trained, by setting the Create trainer mode option.Single Parameter: Choose this option if you already know how you want to configure the model.Parameter Range: Choose this option if you are not sure of the best parameters. Then, specify a range of values and use the Tune Model Hyperparameters module to iterate over the combinations and find the optimal configuration.



1. In Hidden layer specification, select Fully-connected case. This option creates a model using the default neural network architecture, which for a neural network regression model, has these attributes:The network has exactly one hidden layer.The output layer is fully connected to the hidden layer and the hidden layer is fully connected to the input layer.

VII) Evaluate Model

Metrics for regression models

The metrics returned for regression models are generally designed to estimate the amount of error. A model is considered to fit the data well if the difference between observed and predicted values is small. However, looking at the pattern of the residuals (the difference between any one predicted point and its corresponding actual value) can tell you a lot about potential bias in the model. The following metrics are reported for evaluating regression models. When you compare models, they are ranked by the metric you select for evaluation.

- Mean absolute error (MAE) measures how close the predictions are to the actual outcomes; thus, a lower score is better.

- Root mean squared error (RMSE) creates a single value that summarizes the error in the model. By squaring the difference, the metric disregards the difference between over-prediction and under-prediction.
- Relative absolute error (RAE) is the relative absolute difference between expected and actual values; relative because the mean difference is divided by the arithmetic mean.
- Relative squared error (RSE) similarly normalizes the total squared error of the predicted values by dividing by the total squared error of the actual values.Mean Zero One Error (MZOE) indicates whether the prediction was correct or not. In other words: ZeroOneLoss(x,y) = 1 when x!=y; otherwise 0.

Feature	Weight
MSP	0.968419
CROP_APPLE_0	-0.456609
CROP_ONION_4	0.216872
CROP_BAJRI_1	0.132254
CROP_CAPSICUM_2	0.11062
DEMAND	0.0653938
YIELD	-0.0454516
Bias	0.0032481
CROP_GREEN CHILI_3	0.000112084
CROP#unknown__6	0

- Coefficient of determination, often referred to as R^2, represents the predictive power of the model as a value between 0 and 1. Zero means the model is random (explains nothing); 1 means there is a perfect fit. However, caution should be used in interpreting R^2 values, as low values can be entirely normal and high values can be suspect.

Linear Regression:

- 1)Method is Ordinary Least Squares and Online Gradient Descent
- 2)L2 regularisation weight=0.001
- Batch Linear Regressor.

Setting	Value
Bias	True
Regularization	0.001
Allow Unknown Levels	True
Random Number Seed	

Score Model for Linear Regression

CROP	MSP	YIELD	DEMAND	PRICE	Scored Labels
CAPSICUM	-0.700531	-0.434816	-0.423582	-0.479964	-0.572476

Neural Network Regression:

1)Create Train Mode: single parameter or parameter range

2) Hidden Layer Specification: Fully connected Case or custom definition script

3) Number of hidden nodes=100,
Learning rate=0.005,
Number of Learning Iterations=200,
The initial learning weight diameter=0.1
Type of Normaliser=Gaussian Normaliser, Binning Normaliser and MinMax Normaliser

Train Model for Neural Network Regression

Split Data:

1)Splitting mode: Split Rows, Recommender Split, Regular Expression & Regular Derivation

2)Fraction of rows in the first output dataset=0.8

3)Random Seed=(integer Value)

4)Stratified Split:False

Score Model for the Neural Network Regression

CROP	MSP	YIELD	DEMAND	PRICE	Scored Labels
CAPSICUM	-0.108579	-0.442941	-0.428502	-0.11503	-0.039022

Predictive Analysis:

Setting	Value
Is Initialized From String	False
Is Classification	False
Initial Weights Diameter	0.1
Learning Rate	0.005
Loss Function	CrossEntropy
Momentum	0
Neural Network Definition	
Data Normalizer Type	Gaussian
Number Of Input Features	9
Number Of Hidden Nodes	System.Collections

Apply Transformation

CROP	MSP	YIELD	DEMAND	PRICE
ONION	-0.832941	1.538403	1.022296	-0.556704

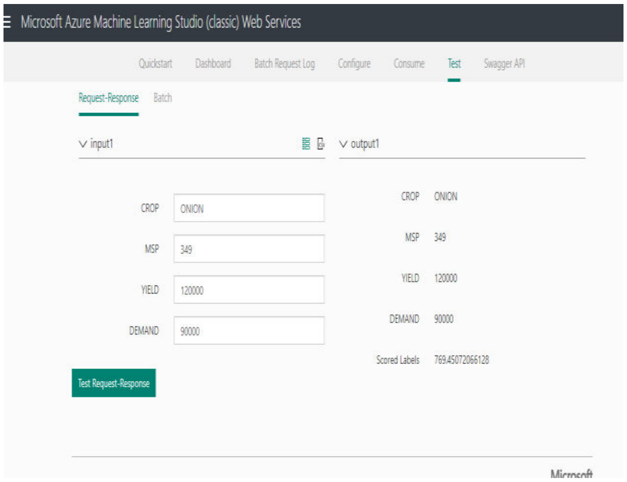
2581-4575
International Journal For Recent Developments in Science & Technology
Crossref
A Peer Reviewed Research Journal

Score Model

CROP	MSP	YIELD	DEMAND	Scored Labels
ONION	-0.832941	1.538403	1.022296	-0.589587

OUTPUT:

Deploy Web Service:



input1

CROP ONION
MSP 349
YIELD 120000
DEMAND 90000

output1

CROP ONION
MSP 349
YIELD 120000
DEMAND 90000
Scored Labels 769.45072066128

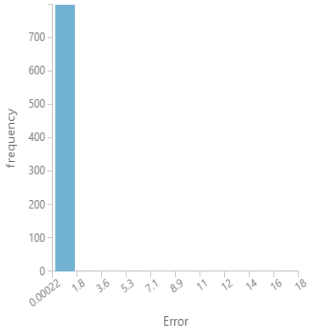
Metrics

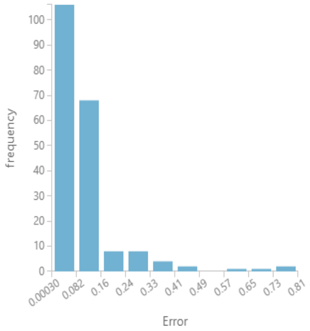
Mean Absolute Error	0.169624
Root Mean Squared Error	0.727286
Relative Absolute Error	0.339277
Relative Squared Error	0.454421
Coefficient of Determination	0.545579

Metrics

Mean Absolute Error	0.108669
Root Mean Squared Error	0.159728
Relative Absolute Error	0.265488
Relative Squared Error	0.074864
Coefficient of Determination	0.925136

00





Enhancement

Crop prediction helps farmers in selecting proper crop for plantation to maximize their earning. Prediction of crops can be accurately done with the help of machine learning techniques and considering the environmental parameters. In future work, the classifiers can be used are support vector machine and artificial hybrid neural networks. Enhancements can be done for the Prediction of crop is done by considering parameters like amount of rainfall, minimum and maximum temperature, soil type, humidity, and soil pH value. The data is collected from the agricultural websites. The data is divided into nine agricultural zones. An interface is been designed through which farmers can enter the required information to predict the crop. Neural network gives 86.80% of prediction accuracy.

References:

[1]Predictive Analysis on Nutritional Disorders in Rice Plant using Regression- *IJCSIT-Volume5 (2), 2014, ISSN: 0975-9646*.

[2] Price Prediction using Linear Regression *https://machinelearningmastery.com/linear-regression-for-machine-earning/*

[3]Panchenko D. 18.443 Statistics for Applications, Section 14, Simple Linear Regression. *Massachusetts Institute of Technology: MIT OpenCourseWare; 2006*.

[4]Elazar JP. Multiple Regression in Behavioral Research: Explanation and Prediction. *2nd ed. New York: Holt, Rinehart and Winston; 1982*.

[5]Mendenhall W, Sincich T. Statistics for Engineering and the Sciences. *3rd ed. New York: Dellen Publishing Co.; 1992*

[6]Chan YH. Biostatistics 201: Linear regression analysis. *Age (years). Singapore Med J 2004;45:55-61*.

[7]Freedman DA. Statistical Models: Theory and Practice. Cambridge, USA: Cambridge University Press; 2009.