UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Buy & Sell Trends Analysis Using Decision Trees
Artículo Académico

Carlos Alejandro Vaca Loaiza

Ingeniería en Sistemas

Trabajo de integración curricular presentado como requisito
para la obtención del título de
Ingeniero de Sistemas

Quito, 07 de mayo de 2020

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE INTEGRACIÓN CURRICULAR

Buy & Sell Trends Analysis Using Decision Trees

Carlos Alejandro Vaca Loaiza

Calificación:

Nombre del profesor, Título académico Noel Pérez, Ph.D

Firma del profesor: ______________________________

Quito, 07 de mayo de 2020

DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante: ____

Nombres y apellidos: Carlos Alejandro Vaca Loaiza

Código: 00124701

Cédula de identidad: 1104162670

Lugar y fecha: Quito, 07 de mayo de 2020

ACLARACIÓN PARA PUBLICACIÓN

UNPUBLISHED DOCUMENT

RESUMEN

Los avances actuales de inteligencia artificial tienden a ser enfocados en técnicas especializadas de deep learning que son computacionalmente caras y requieren una infraestructura costosa. Estas técnicas han mostrado ser particularmente efectivas en ambientes altamente complejos como, procesamiento de imagen, procesamiento de lenguaje natural y la predicción del mercado. Por otra parte, pequeñas compañías están requiriendo más y más acceso a la inteligencia artificial para predecir el comportamiento del cliente y por lo tanto evitar verse afectado por la alta volatilidad y varianza del mercado. Desafortunadamente, la mayoría de estas compañías no son capaces de acceder al costo actual de los métodos avanzados de inteligencia artificial. Por lo tanto, en esta investigación estudiamos una conocida alternativa de bajo costo: árboles de decisión para clasificación. En particular, enfocamos nuestro análisis en los beneficios para analizar las predicciones del mercado con alta exactitud en tres bases de datos: Social Network Advertising Sells, Organic Purchased Indicator, and Online Shoppers Purchasing Intention. Los mejores modelos de árboles de decisiones obtenidos fueron aquellos que produjeron resultados de clasificación entre 93% a 99% de exactitud en predicción. Adicionalmente, se reviso el área bajo la curva y nuestros modelos obtuvieron resultados en el rango de 0.98 a 1.00. Estos resultados muestran que simples modelos como los árboles de decisión son buenos para entender las fluctuaciones y tendencias de los datos del mercado, y dada su simplicidad es una alternativa para las pequeñas compañías dispuestas a utilizar inteligencia artificial.

Palabras clave: Arboles de Decisión, Bases de Datos, Tendencias del Mercado, C4.5, CART.

ABSTRACT

Present artificial intelligence advances tend to be focused on customized deep learning techniques which are computational expensive and require costly infrastructure. These techniques have shown to be particularly effective in highly com- plex environments such as image processing, natural language processing and market price predictions. On the other hand, small companies are requiring more and more access to artificial intelligence to predict customer behavior and hence to avoid to be affected by the highly volatility and variance of the market. Unfortunately, most of these companies may not be able to afford the costs of current artificial intelligence advanced methods. Hence, in this paper we study a low-cost known alternative: decision tree classifiers. In particular, we focus our analysis on the benefits to use them to analyze market predictions with high accuracy over three databases: Social Network Advertising Sells, Organic Purchased Indicator, and Online Shoppers Purchasing Intention. The best decision tree models obtained were those that produced a classification score from 93% to 99% of accurate predictions. In addition, we checked the area under the curve and our models provided results ranging from 0.98 to 1.00. These results show that simple models like decision trees are good to understand the fluctuation and trends from market data, and since its simplicity are an alternative for small businesses willing to try artificial intelligence predictions.

Key words: Decision Tree, Market trend, Databases, C4.5, CART.

AGRADECIMIENTOS

Agradezco a Dios, a mis padres, abuelitos y a mi hermana por su apoyo incondicional, por siempre motivarme a dar mi mejor esfuerzo en todo aspecto de mi vida y a lo largo de toda mi carrera universitaria.

Le doy gracias a Daniel Riofrío y Noel Pérez por motivarme a realizar un aporte a la comunidad, que además me ayudo a crecer como profesional. De la misma forma gracias a todos mis amigos que me apoyaron en el camino.

Carlos

DEDICATORIA

A mis padres, abuelitos, hermana y toda mi familia,

Carlos

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCTION

The uncertainty of the market conveys inherent risks for businesses, specially if those risks are not handled correctly. In fact, market fluctuations and price instability have a direct negative impact to business profits (Bloom, 2009), (Lyu, Cao, Wu & Li, 2020). Therefore, successful organizations prepare to handle risky scenarios based on accounting and optimizing for possible and expected out- comes, respectively (Nooteboom, 2019). In particular, the artificial intelligence (AI) field has contributed with several options that provide consistent results by looking into market features such as those of clients that are not easily affected by the fluctuation of the market (Balter & Pelsser, 2020).

Many classic machine learning techniques have successfully been applied to predict market trends, to enumerate a few, artificial neural networks (ANNs), support vector machines (SVMs), and hybrid models. In particular, ANNs depend of several hyper-parameters to generate a model and the search space of optimal hyper-parameters tend to be a known combinatorial problem. Furthermore, ANNs depend on the quantity of datapoints in the dataset; hence, the bigger the dataset the more accurate the model (Menon, Singh & Parekh, 2019), which undoubtedly demand higher computational resources. SVMs, on the other hand, depend on solving a quadratic optimization problem which usually is computational expensive as the quantity of datapoints in the dataset increases (Duan, Zhu & Lu, 2013). In order to avoid excessive search of hyper-parameters or rely on an expensive optimization problem, other researchers have tried integrating different machine learning techniques with standard hyper-parameters and combining their results, in a majority-vote decision, to provide accurate market predictions (Usmani, Ebrahim, Adil & Raza, 2018).

This work aims to determine the most appropriate decision tree-based classifiers in the classification of the buy-sell market context. Behind this approach, we explore three different entropy indexes to expand the selected DT classifiers. The use of DT models instead of other machine learning classifiers (MLCs) is due to the fact of being faster, less complex, and easier to interpret [8]; which are necessary conditions to take into consideration when using buy-sell market databases. We report the best DT classifier and its best hyper-parameters for each of three market prediction databases.

The remainder of this paper is organized as follows: materials and methods section, presents all experimental steps taken to generate optimum DT classifiers for our three databases. Results and discussion section describe the best models obtained. We use both accuracy and area under the receiver operating characteristic curve (AUC) to support the validity of our results. Finally, conclusions and future work are drawn in the last section.

MATERIALS AND METHODS

This work follows a traditional data mining lifecycle, data collection, data cleaning, feature selection, model preparation, and testing. The following subsections detail particularities of such lifecycle.

Experimental databases

For this study, we collected three different databases based on market predictions. These databases consist of data from: social network advertising sells, organic purchases, and online shoppers purchasing intention. These three databases differ in terms of features and total data points. Nonetheless, they make a unique binary decision. Table I offers a snapshot of each database. Following subsections offer further detail of each database.

Social network advertising sell

This database has been taken from Kaggle (Kaggle, 2020). It contains 401 samples of information about purchased items and their related advertisement. The data was generated in 2017 by Facebook API developers and each sample is composed by categorical features and a single output label. The database includes features such as gender which is of binary nature (male or female), age is a numerical value ranging from 18 to 58 and salary estimates vary from 15,000 to 150,000. Data points are quite sparse in the fourth-dimensional space. The output label represent whether or not the users ended up purchasing the item based on the social media advertisement. Data points that successfully purchased any item correspond to 257 samples and the ones that did not purchased any item are 143.

Organic purchased indicator

This database was also obtained from Kaggle (Kaggle, 2020). This database comes from a supermarket and the task at hand is to determine whether or not a client is likely to buy products based on each client features. This database contains 13 features such as gender, geographic region, loyalty status, affluence grade, among others. The total number of datapoints is 22,000. The output label reveals whether or not a client purchased organic products in the supermarket. There are 4,896 clients that ended up purchasing organic products. The dataset includes data points from January 13th 2019 until the end of that year.

Online shoppers purchasing intention

This database was obtained from UCI Machine Learning Repository (UCI, 2020). This database comes from Google Analytics data compilation over a year. The main objective in this dataset is to determine if a client purchased a product in that web session. This database contains 18 features: 10 numerical and 8 categorical values. There are 12,330 data points, each value represents a specific session. The output label reports if the user wants to buy or not a product in a web site. The database was published in August 13th 2018.

Database	Features	Data Point	Summary
Social Network Advertising Sells	4	401	This database contains client features and whether or not he/she purchased an item based on social media advertisement.
Organic Purchased Indicator	13	22,000	This database contains supermarket client information and whether or not they bought organic products.
Online Shoppers Purchasing Intention	18	12330	Online Shoppers Purchasing Intention database provides online user session's attributes in a web shopping portal and the information of each client to make a purchase.

Table 1Databases used into our model's training and testing

Decision trees classifiers

Decision trees (DT) are supervised machine learning methods used to classify data points according to attributes evaluated by a chosen metric. DTs are constructed from a set of instances following a divide and conquer strategy where if all instances belong to the same class, the tree collapses into a leaf with that specific class as label; otherwise, an attribute is selected to partition all data points according to the chosen metric (Quinlan, 1993). At the top of the tree, a root node is generated from the most general feature of the set, and along its path new nodes are created based on features that become more specific in classifying data points.

In general terms, the creation of a DT is a greedy strategy where local optimal decisions are made while selecting features according to a metric (Esmeir & Markovitch, 2007). This construction demands low computational resources and provides a clear explanation of the decisions made while traversing each node from top to bottom until reaching a leaf (Shamim, Hussain, & Shaikh, 2010). Since, there can be multiple strategies to build DTs and different metrics to select attributes, there exists multiple types of DT models. In this work, we focus on C4.5 and CART (Classification and Regression Trees) models. These DTs models are further analyzed in the following subsections.

C4.5 decision trees

The C4.5 is an algorithm used to create DTs. In particular, this algorithm is an enhancement of its predecessor ID3 (Iterative Dichotomiser 3) invented by Ross Quinlan (Quinlan & Rivest, 1989). As any DT, the construction of this tree depends on selecting the "best" attribute to split a dataset, D into the D_j^{th} partition, at a particular tree node. In fact, at any specificattribute A, C4.5 uses

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

as its attribute selection measure. This measure depends on two concepts, Gain and SplitInfo. Gain over an attribute A is defined as

$$Gain(A) = Info(D) - Info_A(D)$$

and it uses ID3 information measure, i.e., entropy, where p_i is the probability of any data point in D to belong to a category C_i. In addition,

$$Info(D) = \sum_{i=1}^{m} p_i log_2(p_i)$$

$$Info_A(D) = \sum_{i=1}^{m} \left(\frac{D_j}{D} Info(D_j) \right)$$

The term, *SplitInfo* is calculated by

$$SplitInfo_A(D) = -\sum_{j=1}^{n} \frac{|D_j|}{|D|} \times log_2\left(\frac{|D_j|}{|D|}\right)$$

In other words, C4.5 algorithm at every partition step selects an attribute A that provides the best classification based on the amount of information still required to finish the task (Xiaoliang, Hongcan, Jian & Shangzhuo, 2009).

Classification and regression trees (CART)

CART, as its name states, is a tree that can be used either for classification or regression purposes. CARTs are binary trees that use Gini Impurity as its attribute selection measure. The idea behind this measure is that at any node of the tree a decision is made according to the least impurity, i.e., the split, D_j, possesses the best information about dividing categories of a data set D. This measure is calculated by

$$G(D) = 1 - \sum_{i=1}^{m} p_i^2$$

where p_i is the probability of any data point in D to belong to a category C_i. The CART algorithm at every candidate split considers all possible splits in the sequence of values for continuous valued attributes and all possible subset splits for categorical attributes (Sheng & Gengxin, 2010).

Experimental setup

In order to generate DT models, we followed a traditional data mining life-cycle. First, we gathered a set of databases based on our specific research topic, i.e., "market trends prediction". In particular, we selected three databases as men- tioned in section of experimental databases. Then, we preprocessed each database to accommodate each to our model needs. We randomly split the database into a set for training and another for testing. We followed with training different DT models based on several hyper-parameters, and finally, we tested each model for accuracy (ACC) and area under the receiver operating characteristic curve (AUC) in order to establish the validity of our results considering type I and type II classification errors.

This work was implemented using Python 3.7 and its sklearn machine learning API. In order to perform data transformation Pandas library is used (User guide pandas, 2020).

The following subsections explain particular details related to our experimental setup in terms of database preprocessing, splitting criteria for training and testing database partitioning, DT hyper-parameter configuration, and the selection criteria for our best models.

Database preprocessing

Databases usually include different type of attributes, in particular, text-based categorical ones need to be converted to numerical categories and continuous ones need standardization to avoid numerical aberrations and common scale of values while performing calculations. The transformations made on each database are described as follows:

Social network advertising sells database

As part of our preprocessing in this database, we processed the following attributes: gender, age and estimated salary. Gender is a text-based attribute and it was mapped to numeric values 0 and 1. Age and estimated salary, on the other hand, are continuous values and were scaled using standard score which transforms values based on the mean and standard deviation of the attribute set value distribution.

Organic Purchased Indicator

As in the previous database, we processed text-based and continuous values. For text-based values we have the following attributes: gender, geographical region, loyalty status and neighborhood level. Gender was mapped to numeric values in a range of 0 to 2. Geographical region was mapped in a range of 0 to 5. Loyalty status was mapped in a range of 0 to 4. Neighborhood level was mapped in range of 0 to 7. Finally, all other continuous attributes were scaled using standard score as previously explained.

Online Shoppers Purchasing Intention

In this database, we also mapped text-based attributes and scaled continuous ones. Month, visitor type, weekend and revenue are text-based categorical attributes. Month was mapped to numeric values in a range of 0 to 9. Visitor type was mapped in a range of 0 to 2. Weekend was mapped to numeric values 0 and 1. Revenue was mapped to numeric values 0 and 1. Finally, all other continuous attributes were scaled using standard score as previously explained.

Train and test partition

We first split each database selecting data points uniformly random into two partitions: training (80%) and testing (20%). Training and testing partitions are then preprocessed as described in section of databases preprocessing. In order to avoid over fitting on our results, we applied 10-fold cross validation of the training and testing datasets. We used the produced training dataset to fit our DT models under disjoint conditions each time which in turn provides variability avoiding over fitting.

Decision tree configuration

We used a random search 10-fold cross validation algorithm to try out several training hyper-parameters to train our DT models. This method tries random hyper-parameters from a given set or sets of hyper-parameter options and at each step use a discrete subset of hyper-parameters to train a DT performing a 10-fold cross validation over the training set. We report ACC and AUC as an average of all folds for a specific subset of hyperparameter. A brief explanation of each hyper- parameter and its possible values follows:

Criterion

Refers to the attribute selection measure used to select attributes at each node in the DT. Since, we are exploring C4.5 DT and CART, we use entropy based measure as detailed in section of C4.5 decision tree and Gini impurity as detailed in section of classification and regression trees, respectively.

Max depth

Indicates the higher expansion of the tree until the last branch node, also with these we pre prune the tree to avoid over fitting. We tried different values for this hyper-parameter: no limitation of expansion (none), and odd numbers in the range of 3 to 15.

Sample leaf

Refers to minimum number of samples needed to split each internal node. We tried values ranging from 1 to 9.

Splitter

This is the strategy the DT training algorithm chooses an attribute to split the dataset. There are two options: random where at each split decision the algorithm picks the "best" random attribute to split and best the exhaustive best split.

Max features

Indicates the number of features to consider at the split process. We explore three options: none: where the total number of features is considered, sqrt: where the squared root of the total number of features is considered, and log2 where the logarithm base two of the total number of features is considered.

Selection criteria

After hyper-parameter tunning for each database, we select the best classification model based on the best ACC and AUC obtained.

RESULTS AND DISCUSSION

In agreement with our experimental setup (section train and test partitions), 30 different DT models were built and scored under distinct hyper-parameters. Each model is trained and evaluated under 10 different folds during its construction. And, 10 different models are built with respect to each of the databases selected for this study. We chose one model per each database which in turn provided the best average accuracy (ACC) and area under the receiver operating characteristic curve (AUC):

Social network advertising sells

Our best model is a CART using maximum depth of 3, all features as candidates for split, a minimum sample leaf of 8 and exhaustively looking for the best partition. It achieves an average ACC of 93.6% and a AUC of 0.98 over a 10-fold cross validation training scheme.

Organic purchased indicator

Our best model is a CART using maximum depth of 7, all features as candidates for split, a minimum sample leaf of 4 and exhaustively looking for the best partition. It achieves an average ACC of 99.5% and a AUC of 1.00 over a 10-fold cross validation training scheme.

Online shoppers purchasing intention

Our best model is a CART using maximum depth of 9, all features as candidates for split, a minimum sample leaf of 3 and exhaustively looking for the best partition. It achieves an average ACC of 99.8% and a AUC of 1.00 over a 10-fold cross validation training scheme.

Table II summarizes our exploratory results and provides further details of the performance in terms of the selected scores ACC and AUC.

Performance

Our results show that Gini impurity used in CART is a better attribute selection measure for our chosen databases. All of our best models report high ACC and AUC scores using all features as candidates to split the dataset and the strategy to select such split exhaustive "best". This is not a surprise because these hyper-parameters force the DT training algorithm to perform exhaustive computer trials until finding the best possible classification.

Nonetheless, it is interesting to mention that our second best models were trained under least exhaustive search strategies (lower maximum features, higher minimum sample leaf, random split selection) keeping a high ACC (less than 7% difference) and AUC (less than 0.04 difference). Figure 1 and Figure 2 show the best and second-best models for the Organic Purchased Indicator. Figure 1 shows a CART and figure 2 shows a C4.5 that perform equivalently despite morphological and decision differences.

Overall, the implemented hyper-parameter search strategy used in training DT models over our datasets is able to find good results that provide high ACC without lose of generality while classifying new unseen data.

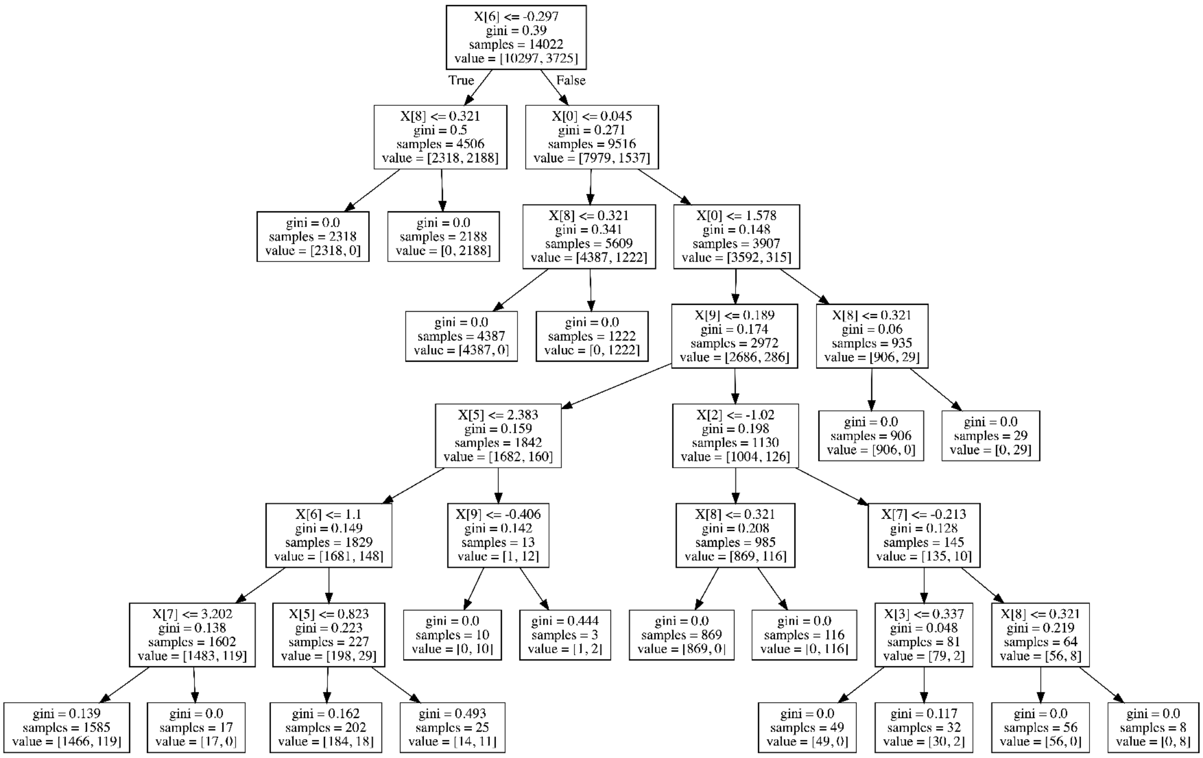

Figure 1 CART for the organic purchased indicator database, configuration number 4 in Table II.

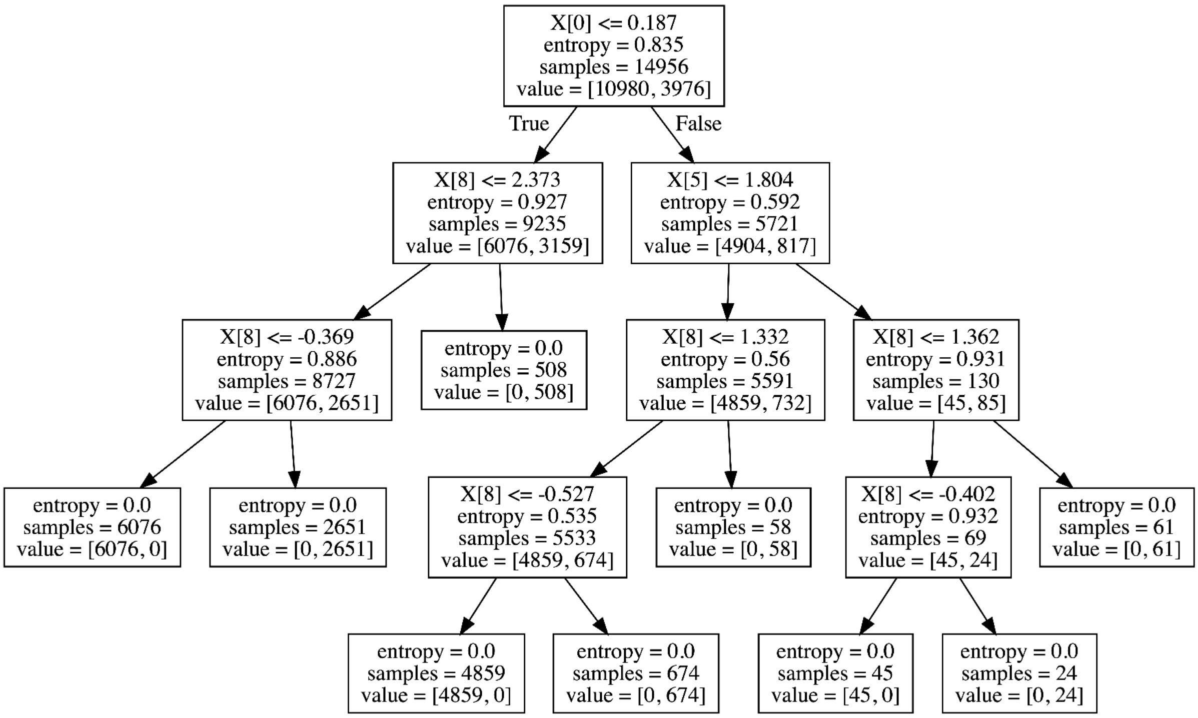

Figure 2 C4.5 tree for the organic purchased indicator database, configuration number 1 in Table II

<table>
<tr><th>Databases</th><th>Config. Number</th><th>Tree Type</th><th>Criteria</th><th>Max Depth</th><th>Max Features</th><th>Min Samples Leaf</th><th>Split</th><th>ACC</th><th>AUC</th></tr>
<tr><td rowspan="10">Social Network Advertising Sells</td><td>1</td><td rowspan="3">C4.5</td><td rowspan="3">entropy</td><td>5</td><td>log_2</td><td>6</td><td>best</td><td>0.889</td><td>0.94</td></tr>
<tr><td>2</td><td rowspan="3">3</td><td>$None$</td><td>2</td><td rowspan="2">random</td><td>0.836</td><td>0.86</td></tr>
<tr><td>3</td><td>log_2</td><td>6</td><td>0.814</td><td>0.96</td></tr>
<tr><td>4</td><td>CART</td><td>gini</td><td>$\boldsymbol{None}$</td><td>8</td><td rowspan="3">best</td><td>0.936</td><td>0.98</td></tr>
<tr><td>5</td><td rowspan="4">C4.5</td><td rowspan="4">entropy</td><td>None</td><td rowspan="2">$sqrt$</td><td>4</td><td>0.835</td><td>0.81</td></tr>
<tr><td>6</td><td>5</td><td>1</td><td>0.863</td><td>0.88</td></tr>
<tr><td>7</td><td>3</td><td rowspan="2">log_2</td><td>4</td><td>random</td><td>0.822</td><td>0.75</td></tr>
<tr><td>8</td><td>5</td><td>5</td><td rowspan="3">best</td><td>0.827</td><td>0.74</td></tr>
<tr><td>9</td><td rowspan="2">CART</td><td rowspan="2">gini</td><td>5</td><td>$None$</td><td>8</td><td>0.861</td><td>0.80</td></tr>
<tr><td>10</td><td>3</td><td>log_2</td><td>5</td><td>0.847</td><td>0.68</td></tr>
<tr><td rowspan="10">Organic Purchased Indicator</td><td>1</td><td>C4.5</td><td>entropy</td><td>9</td><td>$None$</td><td>8</td><td rowspan="3">random</td><td>0.993</td><td>1.0</td></tr>
<tr><td>2</td><td>CART</td><td>gini</td><td rowspan="2">5</td><td rowspan="2">$sqrt$</td><td>3</td><td>0.932</td><td>1.0</td></tr>
<tr><td>3</td><td>C4.5</td><td>entropy</td><td>2</td><td>0.907</td><td>1.0</td></tr>
<tr><td>4</td><td rowspan="3">CART</td><td rowspan="3">gini</td><td>7</td><td>$\boldsymbol{None}$</td><td rowspan="3">4</td><td>best</td><td>0.996</td><td>1.0</td></tr>
<tr><td>5</td><td rowspan="2">3</td><td>log_2</td><td rowspan="2">random</td><td>0.949</td><td>1.0</td></tr>
<tr><td>6</td><td>$sqrt$</td><td>0.923</td><td>1.0</td></tr>
<tr><td>7</td><td>C4.5</td><td>entropy</td><td rowspan="2">None</td><td>$None$</td><td>3</td><td rowspan="2">best</td><td>0.987</td><td>1.0</td></tr>
<tr><td>8</td><td rowspan="2">CART</td><td rowspan="2">gini</td><td>log_2</td><td>8</td><td>0.992</td><td>1.0</td></tr>
<tr><td>9</td><td>3</td><td rowspan="2">$sqrt$</td><td rowspan="2">3</td><td>random</td><td>0.927</td><td>1.0</td></tr>
<tr><td>10</td><td>C4.5</td><td>entropy</td><td>7</td><td>best</td><td>0.951</td><td>1.0</td></tr>
<tr><td rowspan="10">Online Shoppers Purchasing Intention</td><td>1</td><td>C4.5</td><td>entropy</td><td>5</td><td>$sqrt$</td><td>3</td><td rowspan="6">best</td><td>0.920</td><td>1.0</td></tr>
<tr><td>2</td><td>CART</td><td>gini</td><td>7</td><td>log_2</td><td>8</td><td>0.958</td><td>1.0</td></tr>
<tr><td>3</td><td>C4.5</td><td>entropy</td><td rowspan="2">12</td><td rowspan="2">$sqrt$</td><td>3</td><td>0.997</td><td>1.0</td></tr>
<tr><td>4</td><td>CART</td><td>gini</td><td>8</td><td>0.984</td><td>1.0</td></tr>
<tr><td>5</td><td>C4.5</td><td>entropy</td><td>5</td><td>log_2</td><td>5</td><td>0.990</td><td>1.0</td></tr>
<tr><td>6</td><td rowspan="4">CART</td><td rowspan="4">gini</td><td>9</td><td>$\boldsymbol{None}$</td><td>3</td><td>0.998</td><td>1.0</td></tr>
<tr><td>7</td><td>15</td><td>log_2</td><td>15</td><td rowspan="2">random</td><td>0.985</td><td>1.0</td></tr>
<tr><td>8</td><td>7</td><td rowspan="2">$sqrt$</td><td>3</td><td>0.941</td><td>1.0</td></tr>
<tr><td>9</td><td>5</td><td>4</td><td>best</td><td>0.960</td><td>1.0</td></tr>
<tr><td>10</td><td>C4.5</td><td>entropy</td><td>4</td><td>$None7$</td><td>7</td><td>random</td><td>0.990</td><td>1.0</td></tr>
</table>

Table 2 10-Fold cross validation hyper-parameter evaluation over different DT Classifiers for each database in our study. In bold, we show the best set of hyper-parameters with respect to ACC and AUC.

CONCLUSION AND FUTURE WORK

This work shows how to obtain appropriate decision tree-based classifiers in the context of buy-sell market predictions. We have described a sound methodology consisting of database preprocessing, database split, a search strategy to select optimal hyper-parameters and report over a k-fold training method to avoid over fitting.

Data preprocessing is a key step in order to use DT classifiers. Text-based categorical data fields were transformed and mapped into numerical data to facilitate operating on them. In addition, continuous data fields were normalized using standard scaling to avoid numerical aberrations while performing calculations.

Finally, as shown in Figure 1 and Figure 2, decision trees are powerful yet easy to understand machine learning methods. At each node of the tree, a decision is made based on an attribute of the dataset (in Figure 1, first node, parameter six, condition $X[6] \leq -0.297$). That decision leads another decision in a subsequent node which in turn will lead into a conclusion about a data point (classification).

In terms of our future work directions, we envision comparing our DT results against statistical machine learning algorithms such as SVMs and ANNs (deep learning models included) in terms of accuracy, training time and response time. In addition, we plan looking into regression trees to deal with prediction of continuous variables.

REFERENCES

Bloom, N. (2009). The impact of uncertainty shocks. *econometrica*, *77*(3), 623-685.

Lyu, J., Cao, M., Wu, K., & Li, H. (2020). Price volatility in the carbon market in China. *Journal of Cleaner Production*, *255*, 120171.

Nooteboom, B. (2019). Uncertainty and the Economic Need for Trust. In *Trust in Contemporary Society* (pp. 60-73). Brill.

Battro, A., M., Fischer, K.W. & Lena, P.J. (Eds). (2008). *The educated brain: Essays in neuroeducation*. Cambridge, UK: Cambridge University Press.

Balter, A. G., & Pelsser, A. (2020). Pricing and hedging in incomplete markets with model uncertainty. *European Journal of Operational Research*, *282*(3), 911-925.

Berninger, V.W. & Corina, D. (1998). Making cognitive neuroscience educationally relevant: Creating bidirectional collaborations between educational psychology and cognitive neuroscience. *Educational Psychology Review, 10*(3), 343-354.

Menon, A., Singh, S., & Parekh, H. (2019, March). A Review of Stock Market Prediction Using Neural Networks. In *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)* (pp. 1-6). IEEE.

Duan, R. N., Zhu, J. Y., & Lu, B. L. (2013, November). Differential entropy feature for EEG-based emotion classification. In *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)* (pp. 81-84). IEEE.

Usmani, M., Ebrahim, M., Adil, S. H., & Raza, K. (2018, December). Predicting Market Performance with Hybrid Model. In *2018 3rd International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEEST)* (pp. 1-4). IEEE.

Esmeir, S., & Markovitch, S. (2007). Anytime learning of decision trees. *Journal of Machine Learning Research*, *8*(May), 891-933.

“Social network ads — kaggle,” https://www.kaggle.com/rakeshrau/ social- network- ads, (Accessed on 02/03/2020).

“Organics purchase indicator — kaggle,” https://www.kaggle.com/ papercool/organics-purchase-indicator, (Accessed on 02/03/2020).

“Uci machine learning repository: Citation policy,” https://archive.ics.uci.edu/ml/citation policy.html, (Accessed on 03/05/2020).

Shamim, A., Hussain, H., & Shaikh, M. U. (2010, June). A framework for generation of rules from decision tree and decision table. In *2010 International Conference on Information and Emerging Technologies* (pp. 1-6). IEEE.

Berninger, V.W. & Corina, D. (1998). Making cognitive neuroscience educationally relevant: Creating bidirectional collaborations between educational psychology and cognitive neuroscience. *Educational Psychology Review, 10*(3), 343-354.

Xiaoliang, Z., Hongcan, Y., Jian, W., & Shangzhuo, W. (2009, December). Research and application of the improved algorithm C4. 5 on decision tree. In *2009 International Conference on Test and Measurement* (Vol. 2, pp. 184-187). IEEE.

Sheng, B., & Gengxin, S. (2010, August). Data Mining in census data with CART. In *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)* (Vol. 3, pp. V3-260). IEEE.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Layton, R. (2013). API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*.

"User guide — pandas 1.0.1 documentation," https://pandas.pydata.org/ pandas-docs/stable/user guide/index.html, (Accessed on 02/11/2020).

Xiaowei, L. (2014, January). Application of decision tree classification method based on information entropy to web marketing. In *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation* (pp. 121-127). IEEE.

Nugroho, F. S. D., Adji, T. B., & Fauziati, S. (2014, November). Decision support system for stock trading using multiple indicators decision tree. In *2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering* (pp. 291-296). IEEE.

Alawad, W., Zohdy, M., & Debnath, D. (2018, September). Tuning Hyperparameters of Decision Tree Classifiers Using Computationally Efficient Schemes. In *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)* (pp. 168-169). IEEE.

Rathan, K., Sai, S. V., & Manikanta, T. S. (2019, April). Crypto-Currency price prediction using Decision Tree and Regression techniques. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 190-194). IEEE.

Manapragada, C., Webb, G. I., & Salehi, M. (2018, July). Extremely fast decision tree. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1953-1962).

Kim, Y. E., & Yang, H. C. (2020). The effects of perceived satisfaction level of high-involvement product choice attribute of Millennial generation on repurchase intention: Moderating effect of gender difference. *Journal of Asian Finance, Economics and Business*, *7*(1), 131-140.

A. S. Bhatt, "Comparative analysis of attribute selection measures used for attribute selection in decision tree induction," in *2012 International Conference on Radar, Communication and Computing (ICRCC)*, Dec 2012, pp. 230–234.