**Lead Scoring and CTR Prediction:**

**Comparing Machine Learning Algorithms**

Charles Radke,

Graduate Student, Applied Mathematics and Statistics,

Stony Brook University

[charles.radke@stonybrook.edu](mailto:charles.radke@stonybrook.edu)

**Abstract**

*With ample amounts of relevant data, the most successful companies can best leverage this resource for its decision making. Digital marketing is a prime example of this application. Marketing agencies can apply predictive modeling to target online advertisement locations with the highest click-through-rate, a metric directly correlated with an advertisement’s success. Another application of predictive modeling comes with the common practice of creating a sales prioritization of possible clients to focus on, known as lead scoring. The purpose of this paper is to compare the k-nearest neighbor, decision tree and logistic regression classifiers for binary classifying an advertisement being clicked on and a lead converting to a sale. Each algorithm’s performance is measure by its classification accuracy and computation complexity. It was found that for both cases, decision tree was the best performer with both metrics in mind.*

*\*RESULTS\**

1. **Introduction**

As the world becomes increasingly digital, data has been collected at an accelerated pace. This is evident in 2020, with 2.5 quintillion bytes of data generated daily from online activity (Christo, 2021). This plentiful commodity has been increasingly leveraged by almost all industries for its decision making, with 97.2% of organizations investing in artificial intelligence and big data (Christo, 2021).

Within the past decade, marketing agencies have put less of a focus on utilizing offline platforms for their advertising campaigns. Instead, more resources have been directed towards digital outlets, such as web advertisements. When placing an advertisement online, one key metric that is tracked is the ‘click through rate’ (CTR). This rate is calculated by dividing the number of times an advertisement was clicked, divided by the number of visits on that corresponding site. By utilizing big data, an agency can predict a location’s CTR and target accordingly. This task can be thought of as platform targeting.

When someone shows an interest in a product or service, this person is labeled as a ‘lead’. Resources can be used to convert these leads into sales but are often limited. A popular practice is to score leads based on their activity and demographics. After scoring, certain ones are prioritized to allocate resources more efficiently. Big data provides ample historical evidence to identify variables most influential for a lead to convert to a sale. This can be thought of as customer targeting.

This paper will focus on how different machine learning models compare in these two tasks, with one unique data set for each task. Only three models are considered: logistic regression, decision trees, and k-nearest neighbors (KNN). All three models are compared to identify which is most suitable for making accurate predictions. Accurate predictions come at a cost, computation complexity for model fitting and hyperparameter tuning can be troublesome. All these factors are also distinguished.

1. **Literature Review**

Due to the importance of online advertising and sales, research has occurred on which prediction models are best suited for these two tasks. In the paper *“Automating Lead Scoring with Machine Learning: An Experimental Study”*, the authors worked with a similar dataset to compare four different algorithms when predicting lead scoring: logistic regression, decision trees, random forest, and neural networks. However, their dataset consisted of all website visitors, as opposed to just leads. To combat this issue, the data was aggregated in 5 different ways, each having a different interpretation of a lead. Models were t fit on each aggregation and evaluated by their AUC. For all 5 aggregations, the random forest algorithm had the highest AUC value, with varying ranges of bias.

The paper, “Prediction of Lead Conversion with Imbalanced Data”, compares several classifiers’ performances with lead scoring. Like this paper, the author evaluated KNN, logistic regression and decision trees, but also considered support vectors and gradient boosting. Optimal hyperparameters were found by using successive halving grid search and performance was measured by average precision. Interestingly, models were trained with oversampled the conversion observations. Since the corresponding dataset had limited conversions, these observations were randomly duplicated and added to the dataset. Whereas this procedure can cause overfitting, it also allows the model to better identify important features for sales conversion. Combined with oversampling, the gradient boosted classifier resulted in the highest precision score. When comparing the models that this paper focuses on, KNN performed the best, followed by decision trees, and then logistic regression.

In recent years, as opposed to lead scoring, there has been more efforts towards researching click through rate prediction. Blanc (2018) performed a comparative study on this topic with various models, primarily with ensemble methods. As opposed to the oversampling method in Etminan (2021), the author performs under sampling, decreasing the observations of the majority class. In the paper the decision trees, KNN and regularized logistic regression models are compared with the ensemble methods. Among those three algorithms, regularized logistic regression was found to have the highest AUC and accuracy. The KNN model had a higher AUC than the fitted decision tree, but the decision tree had a higher accuracy. It is worth noting that it was found that gradient boosting, random forest, and linear discriminant analysis algorithms had the best performance.

1. **Methodology**

**3.1 Data and Preprocessing**

Two datasets were downloaded from Kaggle. The first dataset pertains to a company that provides paid online courses. When someone is referred or fills out a form on the site, they are considered a lead. The demographics they or their referral include along with their browsing activity provide 56 predictor variables used to predict whether they purchased a course.

Avazu, an online marketing company, released 11 days’ worth of its collected data on Kaggle. The set contains 27 predictor variables, along with a binary response variable that indicates whether the advertisement was clicked or not at a different hour. These predictor variables primarily indicated an observation’s device, website activity and advertisement location/position. This dataset was very large, so a random subset of this data was used to run all models locally.

For both datasets, minimal preprocessing was required. However, all relevant numeric variables were scaled, specifically ones that weren’t factors. This was a requirement for some of the machine learning models to be fit on Skicit Learn. Scaling was also beneficial by improving a model understanding for each numeric variable’s relative relationship.

**3.2 Measurements**

This study evaluated how lead scoring and CTR prediction problems perform with three different algorithms: logistic regression, decision trees, and k-nearest neighbor classifiers. When tuning the models, each algorithm considered variations of two relative hyperparameters. For the tuning times to be compared equally, the number of considered parameter values had to be equal. One hyperparameter value would have four different values, primarily due to the four main logistic regression penalization types. The other hyperparameter would consider thirty possible values. Thirty values were relevant since it was recommended in the CART method for testing maximum depths in a grid search for decision trees (Schratz, Patrick, et al., 2018).

To find optimal hyperparameter pairings, a ten-fold cross validated grid search was implemented. Computationally exhaustive, each possible hyperparameter pairing is fitted and tested in a 10-fold cross validation. After all pairings have been considered, the one that resulted in the highest average cross fold accuracy would be returned. Using cross fold validation for the grid search was ideal since limited variance in using one trainset. The datasets used were not excessively large. Because of this, grid search was not only computationally feasible, but was also an appropriate procedure for comparing processing times.

For the logistic regression tunings, 4 different penalization types were considered, along with 30 different regularization strengths. All combinations used a SAGA solver since it supported all four penalization types on Skicit Learn. For decision trees, a minimum split ranged from 2-5 samples with maximum tree depths ranging from 1-30. The node splitting utilized an entropy criterion, and all other parameters used their default values. Lastly, for the k-nearest neighbor classifier, 1-30 nearest neighbors were considered and distances were calculated with four different metrics: Manhattan, Euclidean, and the Minkowski (power of 3 and 4). For each tuning, the grid search processing times were recorded

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Grid Search 10-Fold CV Accuracy** | | | | | | |
|  | | **Regularization Strength** | | | | |
| 0.0 | 0.009 | 0.008 | … | 1.1 |
| **Penalty** | None |  |  |  |  |  |
| Ridge |  |  |  |  |  |
| Lasso |  |  |  |  |  |
| Elastic |  |  |  |  |  |

Table 1: Grid Search 10-fold CV for Logistic Regression

The average cross validated accuracy was fitting for identifying optimal hyperparameter values. However, it was still necessary to compare each algorithms overfitting tendencies. Lead conversions and CTR’s were predicted with both training and test sets. Accuracies were calculated and compared to pinpoint overfitting inclinations for each task with each model.

1. **Results**

The following section goes over the performance of the three models of interest. The algorithms are compared in their average cross fold accuracy, hyperparameter tuning time, and AUC scores with our test set.

* 1. **Lead Scoring**

Each model varied significantly in its grid search processing time. KNN had the longest processing time, taking almost 40 minutes to process. In the grid search, the model with the highest average accuracy used 30 neighbors with distances calculated with the Manhattan metric. Next, the logistic regression tuning took around 6 minutes and 15 seconds for the model with no penalization had the highest average cross fold accuracy. The optimal decision had a maximum depth of 7 nodes, where a node had to have a minimum of 3 samples to be split.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Lead Scoring Performance** | | | |
| **Method** | | **Parameters** | **Tuning Time** | **Acc.** |
| Logistic Regression | | *C*: None | 377.73 | 0.70 |
| *Reg*.: None |
| Decision Tree | | *Max depth*: 7 | 36.68 | 0.82 |
| *Min split*: *3* |
| KNN | | *K*: *30* | 2330.79 | 0.80 |
| *Metric:* Manhattan |

Table 2: Model Performances for lead scoring

In terms of accuracy, just marginally, the decision tree model had the largest of both predictive measurements, followed by KNN. The logistic regression model was predictor for lead scoring of the three models. Regarding both computation complexity and predictive accuracy, the decision tree was found to be most suitable for this case study.

* 1. **CTR**

Since the data set used for CTR was much larger, the processing times far exceeded the ones used in the lead scoring study. The KNN model still required the largest computational power, with a grid search of lasting almost 26 hours. However, as opposed to lead scoring, the decision tree’s grid search processed for two minutes longer than the logistic regressions. Again, the logistic regression model with no penalization had the highest average cross fold accuracy. As for the decision tree, a maximum depth of 6 paired with a minimum of 3 samples for a split was found to be most optimal. Lastly, the KNN model with 24 nearest neighbors with the Manhattan distance had the highest average accuracy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Lead Scoring Performance** | | | |
| **Method** | | **Parameters** | **Tuning Time** | **Acc.** |
| Logistic Regression | | C: *None* | 508.18 | 0.82 |
| Reg.: *None* |
| Decision Tree | | Max depth: *7* | 632.87 | 0.83 |
| Min split: *3* |
| KNN | | K: 24 | 93482.93 | 0.83 |
| Metric: *Manhattan* |

Table 2: Model Performances for lead scoring

For this lead scoring prediction case study, all three algorithms performed very similarly, with the decision tree and KNN having a slightly higher accuracy.

**Conclusion**

Whereas there were common trends among the models for predicting CTR and lead conversion, each had its own results. For the lead scoring case, the logistic regression model had an extremely poor prediction accuracy and the KNN and decision tree models performed alike. However, not only was the decision tree have the best accuracy, it also was most efficient in its hyperparameter grid search, taking over 30 minutes less to complete. As for CTR, the KNN model and decision tree had the same accuracy, but the decision tree’s grid search was significantly shorter. The KNN model processing time was exuberant, taking over a day to compute with an accuracy of 0.83. This inflated processing time was likely heavily impacted by the large increase in the dataset’s dimensionality.

In comparison to other studies, this papers output has similar results. Nygård and Mezei’s (2020) study found the decision tree to have a higher accuracy than the logistic regression for classifying lead conversion. However, the ensembled method of the decision tree, random forest, was found to be the best performing classifier. When comparing the three algorithms, Etminan (2021) found KNN to have the highest accuracy and AUC, followed by decision trees and then logistic regression. However, the author never touched upon the computational complexity for computing these models, an influential factor for KNN.

Although this paper can be found insightful in differentiating models for these two tasks, it comes with many limitations. For the CTR dataset, the data was unbalanced, influencing model classifications. Oversampling would have made the data set too immense to be run locally. In the future, under sampling methods could be used on these sets to combat both issues. In addition to this, the difference in dataset dimensionality could have an impact on the results, specifically processing times. For example, KNN may not be more computationally complex for lead scoring, rather for more variables. Lastly, more models can be considered. In previous studies, ensemble and boosting methods were found to be very applicable to both tasks.

**References**

Petrov, Christo. “27+ Big Data Statistics - How Big It Actually Is in 2021?” *TechJury*, 1 Nov. 2021, https://techjury.net/blog/big-data-statistics/#gref.

Blanc, Maria do Canto, and Castro Faria. *Click-through rate prediction: a comparative study of ensemble techniques in real-time bidding*. Diss. 2019.

Etminan, Ali. "Prediction of Lead Conversion With Imbalanced Data: A method based on Predictive Lead Scoring." (2021).

Nygård, Robert, and József Mezei. "Automating lead scoring with machine learning: An experimental study." (2020).

Schratz, Patrick, et al. "Performance evaluation and hyperparameter tuning of statistical and machine-learning models using spatial data." *arXiv preprint arXiv:1803.11266* (2018).