PREDICT CUSTOMER SUBSCRIPTIONS BASED ON BANK MARKETING

BIA 678 Big Data Technologies

Tengyue Chen, Guofei Du, Haoxing Zhang

December 9, 2022

Introduction

Marketing campaigns are planned, strategic initiatives to advance a particular business objective, such as increasing consumer awareness of a new product or gathering customer feedback. They often use a variety of media, including but not limited to email, print advertising, television or radio advertising, pay-per-click, and social media, to reach consumers in a variety of ways.

Binary classification refers to classification issues with two class labels. The normal condition is represented by one class in most binary classification issues, and the abnormal condition is represented by the other class.

This project attempts to build a model based on the dataset from bank marketing using binary classification to predict whether the client subscribed to a term deposit.

Data Understanding

We select the dataset from UCI Bank Marketing Data Set (http://archive.ics.uci.edu/ml/datasets/Bank+Marketing#). The Bank Marketing Data Set is a dataset containing information about a direct marketing campaign by a banking institution. The dataset includes information about the customer, such as their age, job, marital status, education, and default status, as well as information about the marketing campaign, such as the contact type, month, and day of week. The dataset includes 20 variables, including both categorical and numeric variables. And The dataset includes 45,211 observations, or individual customer records. Therefore, the dataset could be useful for studying the effectiveness of direct marketing campaigns in the banking industry. The table 1 shows the column of the dataset.

Column	Feature
age	age
job	type of job
marital	marital status

education	education
default	has credit in default?
balance	
housing	has housing loan?
loan	has personal loan?
contact	contact communication type
day	last contact day of the week
duration	last contact duration, in seconds
campaign	number of contacts performed during this campaign and for this client
pdays	number of days that passed by after the client was last contacted from a previous campaign
previous	number of contacts performed before this campaign and for this client
poutcome	outcome of the previous marketing campaign
deposit	has the client subscribed a term deposit?

Table1. Feature of Columns in dataset

EDA

After knowing the basic information of the dataset, we use Exploratory Data Analysis (EDA) to better understand the data. Figure 1 shows the distribution about the age. Most of the people in the data set are between 30 and 40 years old. Figure 2 shows the amount about the marital status.

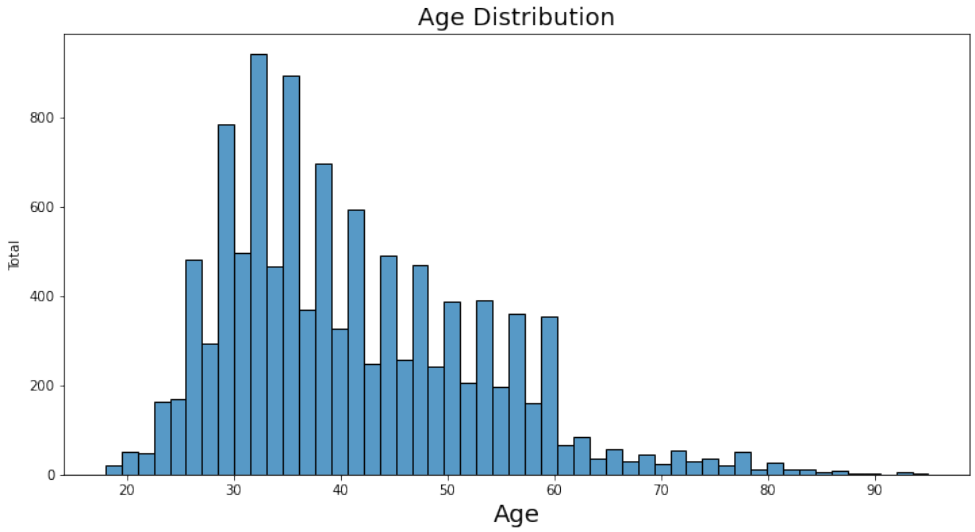

Figure 1. Age Distribution

Figure 2 shows that the distribution about the job. Management is the job with the most people.

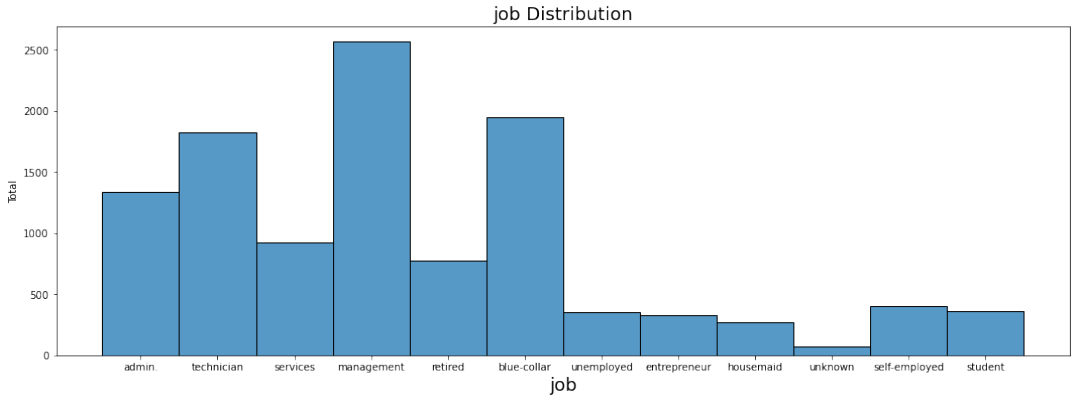

Figure 2. Job Distribution

Afterwards, we combined education and average age to understand the data. Average age decreases as education attainment increases (Figure 3).

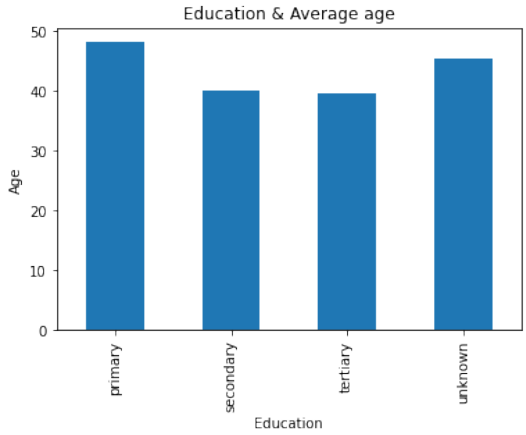

Figure 3. Education & average age

We use boxplot to observe outliers in the dataset. There are obvious abnormalities in the balance in the data set (See Figure 4). Therefore, we analyze the distribution of balance. The distribution of balance is concentrated around 0 (see Figure 5). In addition, because there is no

description of balance in the data set, we delete balance.

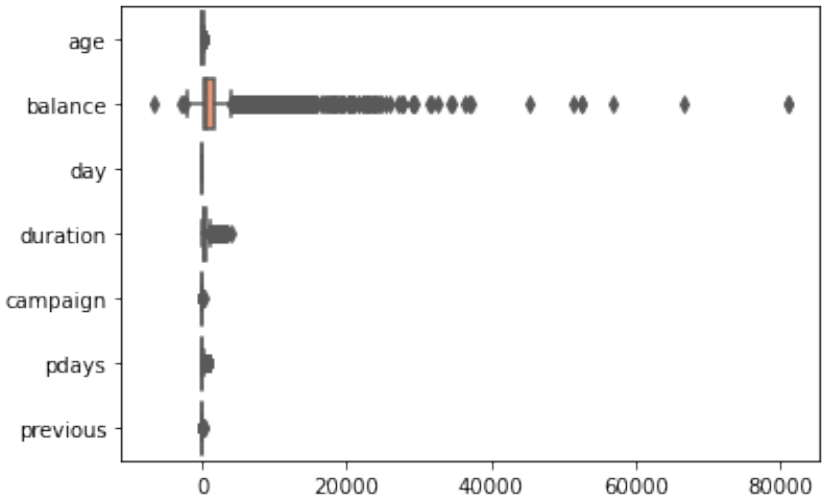

Figure 4. boxplot of dataset

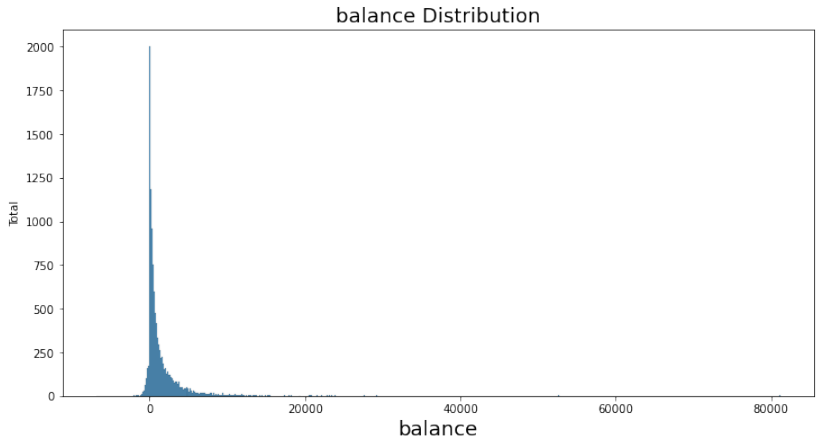

Figure 5. balance distribution

Figure 6 shows the results of Bi-Variate Analysis. Using the Bi-Variate Analysis to understand the relationship between variables. And Figure 8 shows the correlation between the variables.

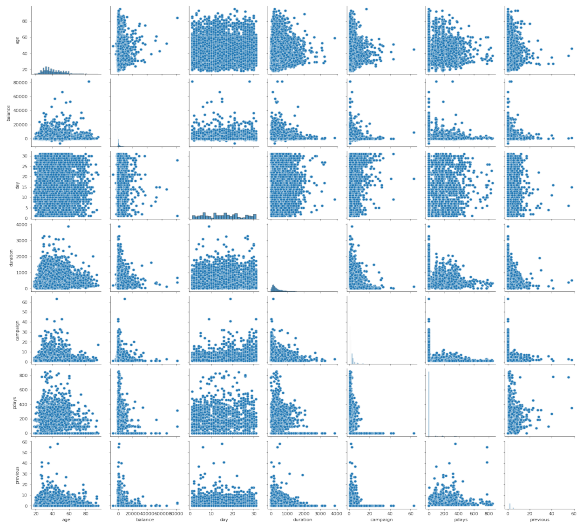

Figure 6. Bi-Variate Analysis Result

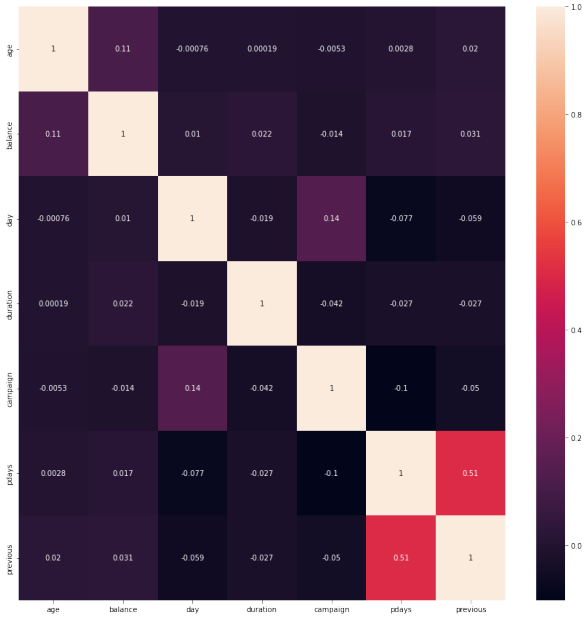

Figure 7. Correlation

Methodology

This project use PySpark on AWS. Amazon Web Services (AWS) is a cloud computing platform that provides a variety of services, including computing, storage, and analytics

(Amazon Web Services, 2022). PySpark is a Python library that allows users to easily and interactively work with large datasets using the Apache Spark distributed computing platform. Apache Spark is an open-source data processing framework that is designed for large-scale data processing (Apache Spark, 2018).

Modeling

In this project, two model are used. First, a decision tree is a type of machine learning model that uses a tree-like structure to make predictions based on the values of multiple attributes (Breiman et al., 2017). Decision tree models are constructed by dividing the data into smaller and smaller subsets, called nodes, based on the values of one or more attributes. At each node, the model decides based on the attribute with the highest information gain, which is a measure of the improvement in the model's accuracy that results from splitting the data on that attribute (Breiman et al., 2017).

The other model we used is Gradient-boosted trees. Gradient-boosted trees (GBTs) are a type of supervised learning algorithm that can be used for classification and regression (Friedman, 2001). GBTs work by training multiple decision trees on the data, and using the errors made by the first tree to train the second tree and next tree; therefore, allows the trees to learn from the mistakes made by previous trees, and results in a model that is more accurate than a single decision tree (Friedman, 2001).

To build the two model, we use the StringIndexer, OneHotEncoderEstimaror, and other function to fit the dataset. And the dataset split to a train set and a test set. Figure 9 shows the Decision Tree model. Using the ParamGridBuilder to find the best model. Figure 10 show the GBTs model. And we use the ParamGridBuilder to build the best model.

```
[9]: from pyspark.ml.classification import DecisionTreeClassifier
     Last executed at 2022-12-08 03:12:27 in 64ms

[10]: dt = DecisionTreeClassifier(featuresCol = 'features', labelCol = 'label', maxDepth = 3)
      model_dt = dt.fit(training)
      pred_dt = model_dt.transform(test)
      Last executed at 2022-12-08 03:12:34 in 7.37s
      ▸ Spark Job Progress

[11]: from pyspark.ml.evaluation import RegressionEvaluator
      evaluator = RegressionEvaluator()
      rmse_dt = evaluator.evaluate(pred_dt,{evaluator.metricName: "rmse"})
      rmse_dt
      Last executed at 2022-12-08 03:12:36 in 1.64s
      ▸ Spark Job Progress
      0.4916988552946748

[12]: from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
      param_grid_dt = ParamGridBuilder()\
                  .addGrid(dt.maxDepth, [1,2,3,4,5])\
                  .build()
      crossvalidate = CrossValidator(estimator=dt,estimatorParamMaps=param_grid_dt, evaluator=evaluator,numFolds=5)
      tuned_model_dt = crossvalidate.fit(training)
      Last executed at 2022-12-08 03:13:20 in 44.48s
      ▸ Spark Job Progress

[13]: model_dt_best = tuned_model_dt.bestModel
```

Figure 9. Decision Tree model

```
[17]: from pyspark.ml.classification import GBTClassifier
      Last executed at 2022-12-08 03:13:26 in 318ms

[18]: gbt = GBTClassifier(maxIter=5, maxDepth=2)
      model_gbt = gbt.fit(training)
      pred_gbt = model_gbt.transform(test)
      Last executed at 2022-12-08 03:13:32 in 5.60s
      ▸ Spark Job Progress

[19]: rmse_gbt = evaluator.evaluate(pred_gbt,{evaluator.metricName: "rmse"})
      rmse_gbt
      Last executed at 2022-12-08 03:13:33 in 1.10s
      ▸ Spark Job Progress
      0.4925792478623212

[20]: param_grid_gbt = ParamGridBuilder()\
                  .addGrid(gbt.maxIter, [1,2,3,4,5])\
                  .addGrid(gbt.maxDepth, [1,2,3,4,5])\
                  .build()
      crossvalidate = CrossValidator(estimator=gbt,estimatorParamMaps=param_grid_gbt, evaluator=evaluator,numFolds=5)
      tuned_model_gbt = crossvalidate.fit(training)
      Last executed at 2022-12-08 03:16:50 in 3m 17.28s
```

Figure 10. GBTs model

Result

After we tuned the two models' parameter, such as the max depth of the decision tree and the max iteration of gradient-boosted tree, we got the best performance of these two models. In the

next step we tested the error of our models and compared the results from them. Then we tried to use less data to measure the impact of scale for these two models respectively. In order to test the error of models, we decided to use root mean square error (RMSE), area under the ROC curve (AUC-ROC) and area under the precision-recall curve (AUC-PR). Then we also compare the running time of models with different scale of data. However, since the size of our data is not big enough, there is not an obvious difference of running time.

- Root mean square error (RMSE): common measure of the difference between the sample value predicted by a model and the observed value.
- Area under the ROC curve (AUC-ROC): metric to calculate the overall performance of a classification model based on area under the ROC curve.
- Area under the precision-recall curve (AUC-PR): model performance metric for binary responses that is appropriate for rare events and not dependent on model specificity.

	Decision Tree Classifier	**Gradient-Boosted Trees Classifier**
RMSE	0.458	0.452
AUC-ROC	0.698	0.874
AUC-PR	0.712	0.829

Table 2. Results of best models

To test the error, we used the 80% of data to train both models and used 20% of data to test, and the above Table 2 shows the result of the two models. For AUC-ROC and AUC-PR, the GBTs Classifier has higher value, so it performs better than the Decision Tree Classifier. For RMSE, there is not an obvious difference between these two models. In conclusion, the GBTs Classifier performs better.

To compare the impact of different scales, we tried to use 60% and 80% of data to train the model and 20% of data to test the model. The Table 3 shows the results of different scales for

these two models. It is observed that for both the two Classifiers though the accuracy dips a little bit for 80% of the data size, it almost remains the same for them.

DT Classifier	**60% of data**	**80% of data**
RMSE	0.477	0.458
AUC-ROC	0.746	0.698
AUC-PR	0.720	0.712
GBTs Classifier	**60% of data**	**80% of data**
RMSE	0.478	0.452
AUC-ROC	0.892	0.874
AUC-PR	0.829	0.829

Table 3. Results of different scales

Conclusion

From the EDA part, the data was effectively analyzed and visualized. Future research is possible because the models may be tested on different platforms, like Google Platform, which could show us different runtime and scalability findings. After modeling and comparing the results between two different algorithms, the Gradient-Boosted Trees Classifier shows a better performance on this job, and this model has lower RMSE and higher AUC score, which means the model can make less error during the binary classification. For tuning the parameters, we used Grid Search to find the best parameters for the algorithms, which is an essential way to reduce the error and improve the efficiency of the model. Adding more data is also crucial in the future work, since the current size of data is severely unbalanced, larger data would be quite advantageous for us. After measuring the impact of scale, it can be concluded that our model can be applied to larger datasets, and if our model can be run on cloud computing services with parallel computing, it will create more accurate predictions with less run time.

Reference:

Amazon. (2022). *Amazon Web Services*. Retrieved December 8, 2022, from https://aws.amazon.com

Apache spark. (2018). *Apache spark- unified engine for large-scale data analytics*. Apache Spark - Unified Engine for large-scale data analytics. Retrieved December 8, 2022, from https://spark.apache.org/

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). Classification and regression trees. https://doi.org/10.1201/9781315139470

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232.

Moro, S., Laureano, R., & Cortez, P. (2014, June). UCI Machine Learning Repository: Bank Marketing Data Set. Retrieved December 9, 2022, from https://archive.ics.uci.edu/ml/datasets/Bank%20Marketing