UCD Michael Smurfit
Graduate Business School

Classification Use Case Report

User Purchasing Intent Prediction

by
Chen Chen (Student ID: 21202636)
Xiuqi Du (Student ID:20211717)
Yilan Li (Student ID: 21200016)

UCD MIS41270
Data Management and Mining
Elayne Ruane
2022

Table of Contents

1. Introduction **3**
2. Descriptions of the Use Case **3**
2.1 Company Details 3
2.2 Description of the Problem 3
2.3 Solutions 3
2.4 Result 3
3. Descriptions of the Dataset **3**
3.1 Data Source 3
3.2 Dataset Size 4
3.3 Key Attributes 4
3.4 Limitations of the Dataset 5
3.5 General Data Statistics 6
4. Pre-processing of the Data **7**
4.1 Data Cleaning 7
4.2 Augmentations on the Data 7
4.2.1 Feature Selection 7
4.2.2 SMOTE oversampling 10
4.3 Normalization of Numerical Variables 11
4.4 Transformation of Categorical Variables to Dummy Variables 11
5. Descriptions of Classification Models **12**
5.1 Naive Bayes 12
5.2 Decision Tree 13
5.3 Random Forest 14
5.4 Logistic Regression 15
6. Implementation of Classification Algorithms **16**
7. Comparison of Classification Results **16**
8. Recommendations **20**
9. Ethics **22**
10. Conclusion **22**
References **22**

1. Introduction

A large retail company has 100 brick and mortars over the country, and they also have an online business division that sells the same products but with online special offers. The company's administrations are interested in the recent development of AI technology, and they turned to us for consulting and tried to find out whether they should introduce similar technologies to help their businesses grow or reduce their costs. Since we cannot get access to the company's operation data, and the time is limited for us to gather enough information with legal approval, we decided to present our client with a classification use case. This use case can showcase how artificial intelligence can be applied in similar business field and what values can it bring to the company.

2. Descriptions of the Use Case

2.1 Company Details

The company in the use case sell office supplies online, and they have about 50 employees all together: one general manager with seven division managers. Their divisions include sales, procurement, administration, accounting, HR, design and quality control.

2.2 Description of the Problem

The company has a thriving online business, but they would like to achieve higher conversion rate by analysing the user behaviours. They would like to know whether the users who viewed their websites have the intention to purchase their products, and if so, they plan to convince those users by sending them unique content that introduces the product in details.

2.3 Solutions

We built several classification models to predict whether each user has purchasing intent or not based on their online behaviours. And we chose the best model for prediction. For users classified as with purchasing intent, this company can target these people for marketing, including providing unique content, offers, and advertisements.

2.4 Result

We are able to identify our targeted users with about 90% recall and accuracy and we can send them our product information to give them a nudge to purchase our products. Thus, the overall conversion rate will be increased and the company's revenue will be higher.

3. Descriptions of the Dataset

3.1 Data Source

Since we were unable to obtain data from the company, we utilized a dataset that could be used to analyse customer purchasing intent. This dataset was obtained from publicly available online sources (https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Data set) and we thought it would be a good fit to reflect the company's online business needs and development. The values in this dataset are derived from the URL information of the pages visited by users and page metrics measured by "Google Analytics".

3.2 Dataset Size

The dataset consists of feature vectors belonging to 12,330 sessions. The dataset was formed so that each session would belong to a different user in a 1-year period to avoid any tendency to a specific campaign, special day, user profile, or period.

3.3 Key Attributes

The dataset consists of 10 numerical and 8 categorical attributes.

Top 5 rows in the dataset:

	Administrative	Administrative_Duration	Informational	\
0	0	0.0	0	
1	0	0.0	0	
2	0	0.0	0	
3	0	0.0	0	
4	0	0.0	0	

	Informational_Duration	ProductRelated	ProductRelated_Duration	\
0	0.0	1	0.000000	
1	0.0	2	64.000000	
2	0.0	1	0.000000	
3	0.0	2	2.666667	
4	0.0	10	627.500000	

	BounceRates	ExitRates	PageValues	SpecialDay	Month	OperatingSystems	\
0	0.20	0.20	0.0	0.0	Feb	1	
1	0.00	0.10	0.0	0.0	Feb	2	
2	0.20	0.20	0.0	0.0	Feb	4	
3	0.05	0.14	0.0	0.0	Feb	3	
4	0.02	0.05	0.0	0.0	Feb	3	

	Browser	Region	TrafficType	VisitorType	Weekend	Revenue
0	1	1	1	Returning_Visitor	False	False
1	2	1	2	Returning_Visitor	False	False
2	1	9	3	Returning_Visitor	False	False
3	2	2	4	Returning_Visitor	False	False
4	3	1	4	Returning_Visitor	True	False

Numerical Attributes

1. Administrative: Number of pages visited by the visitor about account management;
2. Administrative Duration: Total amount of time (in seconds) spent by the visitor on account management related pages;
3. Informational: Number of pages visited by the visitor about Web site, communication and address information of the shopping site;
4. Informational Duration: Total amount of time (in seconds) spent by the visitor on informational pages;
5. Product Related: Number of pages visited by visitor about product related pages;
6. Product Related Duration: Total amount of time (in seconds) spent by the visitor on product related pages;
7. Bounce rate: Average bounce rate value of the pages visited by the visitor;
8. Exit Rate: Average exit rate value of the pages visited by the visitor;
9. Page Value: Average page value of the pages visited by the visitor;
10. Special Day: Closeness of the site visiting time to a special day.

Categorical Attributes

1. Operating Systems: Operating system of the visitor;
2. Browser: Browser of the visitor;
3. Region: Geographic region from which the session has been started by the visitor;
4. Traffic Type: Traffic source by which the visitor has arrived at the Web site (e.g., banner, SMS, direct);
5. Visitor Type: Visitor type as ‘‘New Visitor,’’ ‘‘Returning Visitor,’’ and ‘‘Other’’;
6. Weekend: Boolean value indicating whether the date of the visit is weekend;
7. Month: Month value of the visit date;
8. Revenue: Class label indicating whether the visit has been finalized with a transaction.

No. 1 to 6 attributes in numerical attributes indicate the number of different types of pages viewed by the user during that visit, and the total time spent viewing each page category. The values of these attributes are derived from the URL information of the pages the user visits, and are updated in real time when the user takes some action, such as moving from one page to another.

No. 7 to 10 attributes in numerical attributes can be stored in the application database of all the web pages of the e-commerce site of the developed system and are automatically updated on a regular basis. The value of the " Bounce Rate" of a web page is the percentage of visitors who access the site from that page and leave without initiating any other requests to the analytics server during their visits. The value of the "Exit Rate" for a particular web page is calculated as a percentage of all page views for that web page and is the last time in that visit. The "Page Value" represents the average value of the web pages accessed before the user completes a transaction. The "Special Days" indicates how close the site visit time is to certain special days when login periods are most likely to end with a transaction. The value of this attribute is determined by taking into account e-commerce dynamics, such as the time between the order date and the delivery date. For example, for Valentine's Day, this value takes a non-zero value from February 2 to February 12 and zero before and after this date. However, the maximum value is 1 February, unless it is close to another special date, which is 8 days (Sakar et al., 2019).

The dataset also includes operating system, browser, region, traffic type, visitor type as returning or new visitor, a Boolean value indicating whether the date of the visit is weekend, and month of the year.

The “Revenue” attribute is used as the class label. It’s the purchasing intent of the visitor using aggregated pageview data kept track during the visit along with some session and user information. Purchasing intent refers to the users who eventually complete the purchase to generate revenue, i.e., all users who have consumed regardless of the amount are classified as users with purchasing intent, and those who have not consumed are classified as users without purchasing intent.

3.4 Limitations of the Dataset

This data set mainly contains data about users' web browsing behaviours, which is used to make analysis. However, there are some attributes and characteristics of users that

can also have a significant impact on their purchasing intention and behaviours. For example, the age and gender of the user, for a company that sells building materials in brick and mortars, the proportion of middle-aged and male customers may be higher. Others factors like what kind of job the customers are doing, how much they make per year, whether they are married, etc will also contribute to the accuracy of the classification models. If these factors are gathered and taken into account in the classification, then the classification model will be more useful for the company's business. The dataset we use lacks these attributes, so there is some limitation.

3.5 General Data Statistics

We read the csv file of the dataset into a pandas DataFrame and used the describe function to display the descriptive statistics of the numerical attributes in the dataset (for a complete presentation, we transposed the rows and columns):

Attributes	count	mean	std	min	25%	50%	75%	max
Administrative	12330	2.315166	3.321784	0	0	1	4	27
Administrative_Duration	12330	80.81861	176.7791	0	0	7.5	93.25625	3398.75
Informational	12330	0.503569	1.270156	0	0	0	0	24
Informational_Duration	12330	34.4724	140.7493	0	0	0	0	2549.375
ProductRelated	12330	31.73147	44.4755	0	7	18	38	705
ProductRelated_Duration	12330	1194.746	1913.669	0	184.1375	598.9369	1464.157	63973.52
BounceRates	12330	0.022191	0.048488	0	0	0.003112	0.016813	0.2
ExitRates	12330	0.043073	0.048597	0	0.014286	0.025156	0.05	0.2
PageValues	12330	5.889258	18.56844	0	0	0	0	361.7637
SpecialDay	12330	0.061427	0.198917	0	0	0	0	1
OperatingSystems	12330	2.124006	0.911325	1	2	2	3	8
Browser	12330	2.357097	1.717277	1	2	2	2	13
Region	12330	3.147364	2.401591	1	1	3	4	9
TrafficType	12330	4.069586	4.025169	1	2	2	4	20

And we also found the unique values of categorical variables using the unique function:

Attributes	number of unique values	values
Operating Systems	8	1, 2, 3, 4, 5, 6, 7, 8
Browser	13	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
Region	9	1, 2, 3, 4, 5, 6, 7, 8, 9
Traffic Type	20	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
Visitor Type	3	'Returning_Visitor', 'New_Visitor', 'Other'
Weekend	2	True, False
Month	10	'Feb', 'Mar', 'May', 'Oct', 'June', 'Jul', 'Aug', 'Nov', 'Sep', 'Dec'
Revenue	2	True, False

4. Pre-processing of the Data

4.1 Data Cleaning

We checked the dataset for missing values and formatting issues, but there are none. The dataset is pretty clean, so we don't have much to do in this step. When we checked for repeated records, there are 125 records with the same values. However, we know that each session belongs to a specific user, so these records shall not be treated as repeated records. As for outliers, there are many attributes which have values outside their IQR, but these values are the aggregated numbers within one year. The numbers can vary a lot between each user, so these values shall not be treated as outliers. The only important part we did in this step is to transform the data types.

Six columns have been transformed into categorical datatypes:

```
"""
df["OperatingSystems"] =
df["OperatingSystems"].astype("category")
df["Browser"] = df["Browser"].astype("category")
df["Region"] = df["Region"].astype("category")
df["TrafficType"] = df["TrafficType"].astype("category")
df["Weekend"] = df["Weekend"].astype("category")
df["Revenue"] = df["Revenue"].astype("category")
"""
```

4.2 Augmentations on the Data

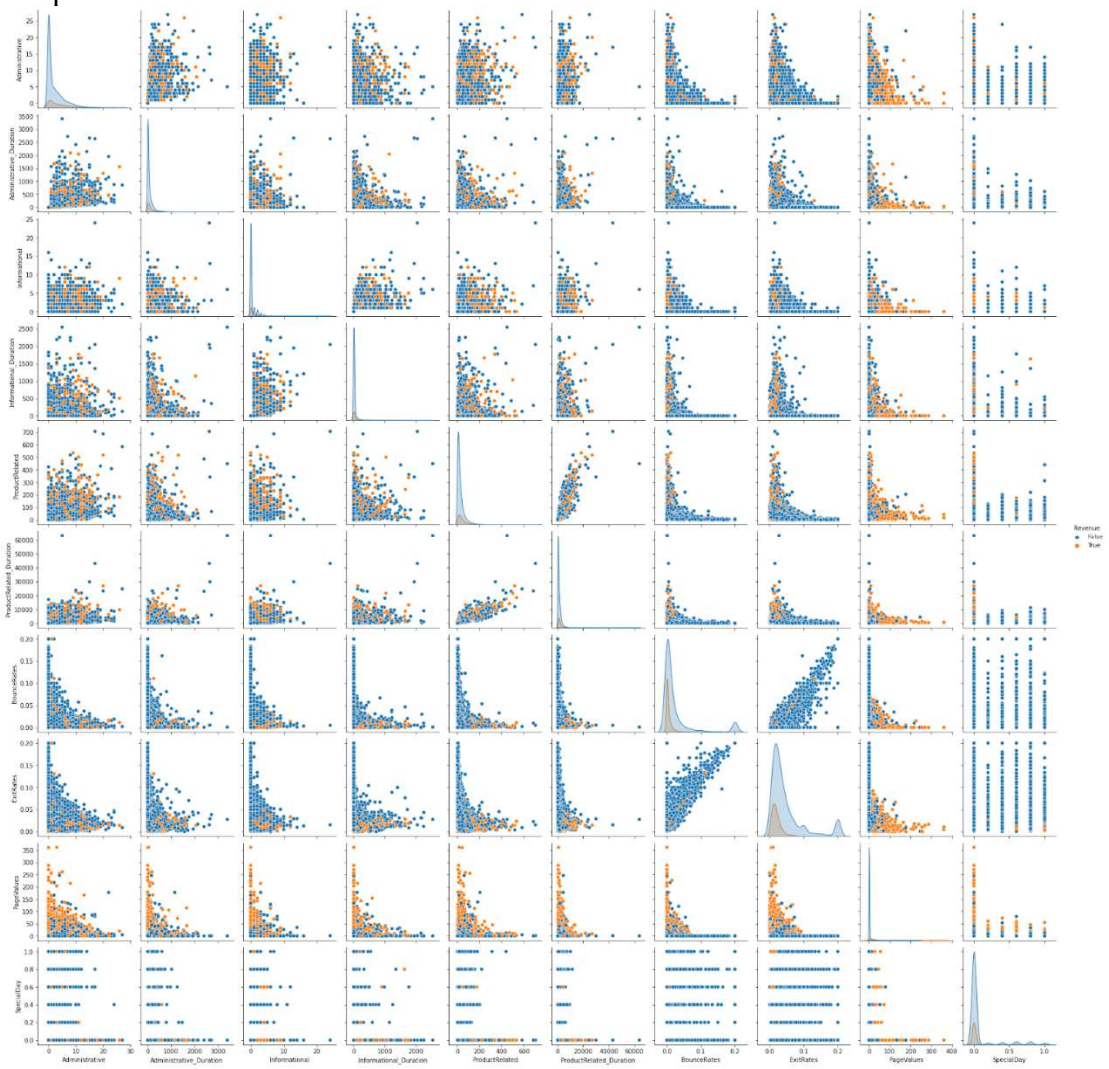
4.2.1 Feature Selection

We applied feature selection techniques to improve the performance of the classification models. And we investigated whether better or similar classification performance can be achieved with fewer numbers of features or some features combined together. We decided to apply the filter-based feature selection instead of wrapper algorithms, and we used the correlation coefficient (Pearson's r), mutual information (MI) filters in our experiments.

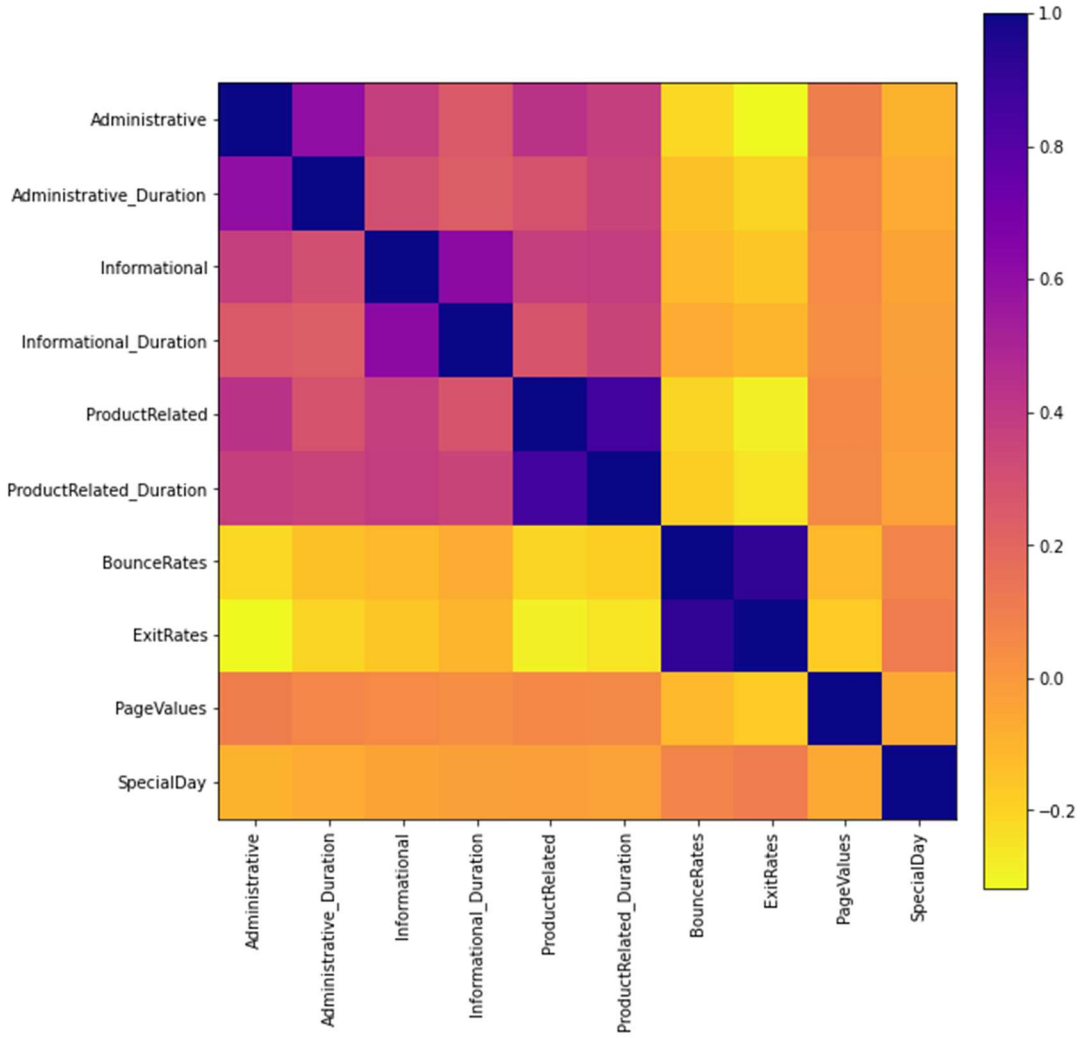
Notes: Since the correlation coefficient can only capture linear relations, we used MI to also capture nonlinear relations.

First, we drew the pairplot between each numerical attribute pair (pairplot graph shown below), and calculated the correlations between these pairs and drew the heatmap to show how strongly they are related with each other (heatmap shown below). We found that 2 pairs of attributes are strongly correlated with each other: "ProductRelated" & "ProductRelated_Duration" (r=0.86), "BounceRates" & "ExitRates" (r=0.913). And we decided to combine these two pairs instead of removing one of each because we couldn't know which one is not important. As a result, the model performance increased slightly.

Pairplot:


Administrative
Administrative_Duration
Informational
Informational_Duration
ProductRelated
ProductRelated_Duration
BounceRates
ExitRates
PageValues
SpecialDay
Revenue
False
True

Heatmap:



And then, we used the SelectKBest function from sklearn.feature_selection along with mutual_info_classif to select the top 20 features that are connected most closely to the class label. And we found that 3 features - "SpecialDay", "TrafficType", "Weekend" - are not so important, so we dropped these three columns of data. As a result, the model performance didn't decrease.

Code for mutual information feature selection:

```
"""
selector = SelectKBest(mutual_info_classif, k=20)
selector.fit(X, y)
cols = selector.get_support(indices=True)
features_new = df.columns[cols]
print(features_new)
"""
```

More information on the correlation coefficient and mutual information:

Pearson's r:
Pearson's r is a measure of linear correlation between two variables. The value of r is between -1 and 1. When r=1, it means that the two variables are completely positive related; when r=-1, it means that the two variables are completely negative related; when r=0, it means that the two variables are not linearly correlated at all. R is calculated as following:

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}}.$$

Mutual Information:
MI is a measure of mutual dependence of the two variables. From the perspective of information gain, MI is the uncertain decreased amount in y due to X. The larger the value of MI, the stronger correlation is between the independent variable and the dependent variable. MI is calculated as following:

$$I(X;Y) = \sum_{x \epsilon X}\sum_{y \epsilon Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}$$

4.2.2 SMOTE oversampling

Of the 12,330 sessions in the dataset, 84.5% (10,422) were negative class samples that did not end with shopping, and the rest (1908) were positive class samples ending with shopping. Negative records are way more than positive records.

This imbalanced dataset can lead to a big problem, since the classification models are built to achieve their best performance, these models are most likely to classify the unseen data points to the majority class. In such cases, the models cannot generalize the patterns in the dataset well, and will become useless models.

We applied the SMOTE oversampling technique to make more records of the positive classes to get an even distribution of the two classes to solve this issue, and we imported the imbalanced-learn module to do it.

As a reminder, the SMOTE oversampling technique shall only be applied to the training set instead of the whole dataset, because when we validate or test the model performance, we need to make sure that the validation set and test set are similar to the real scenarios.

Code for applying SMOTE on train set:

```
"""
smt = SMOTE()
X_train, y_train = smt.fit_resample(X_train, y_train)
"""
```

More information on SMOTE:

SMOTE:
Class imbalance is a scenario that arises when we have an unequal distribution of class in a dataset i.e. the no. of data points in the negative class (majority class) is very large compared to that of the positive class (minority class). And this problem can be solved using SMOTE.

SMOTE is called the Synthetic Minority Oversampling Technique, and this technique works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

4.3 Normalization of Numerical Variables

We normalized the 7 numerical variables using the robust normalization techniques, because we found that there are outliers in the dataset, and we used the RobustScaler from scikit-learn to do it.

Code for normalization:

```
"""
transformer = RobustScaler().fit(X_train[col2])
X_numerical_train =
pd.DataFrame(transformer.transform(X_train[col2]), \
columns=["Administrative","Administrative_Duration","Info
rmational","Informational_Duration","ProductRelated^Durat
ion","Bounce^Exit","PageValues"])
"""
```

As a reminder, normalization shall be performed on the training set first, and then we can apply the same scale on the validation set and test set. This is to avoid bringing any information of the validation set and test set into the model building phase.

4.4 Transformation of Categorical Variables to Dummy Variables

We transformed the 6 categorical variables to dummy variables, because categorical values need to be represented as vectors in the dimensional space, and we used the pandas get_dummies function to do it.

Code for dummy variable:

```
"""
col1 =
df[["Month","OperatingSystems","Browser","Region","Visito
rType","Revenue"]]

categorical = pd.get_dummies(col1, drop_first=True)
"""
```

As a reminder, categorical variables shall not be transformed into numerical values, since categorical variables cannot be compared with each other. And it will be meaningless and wrong to represent them with certain numerical values.

5. Descriptions of Classification Models

5.1 Naive Bayes

The Naive Bayes method is a classification method based on the Bayes' Theorem and the assumption of conditional independence of features, where naive refers to conditional independence. Instead of returning the classification label directly, Naive Bayes returns the probability of belonging to a certain classification label when classifying. For example, to determine the category of an article. The probability of each article belonging to a certain category is calculated, and which category accounts for a larger percentage of the articles is classified as which category. In simple terms, Naive Bayes is a classification algorithm based on the magnitude of the probability.

Three different types of Naive Bayes model algorithms are provided in scikit-learn. For example, Gaussian Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, etc. In our case, we used Gaussian Naive Bayes.

The Gaussian distribution, also known as the normal distribution, is a statistical model that describes the statistical distribution of continuous random variables in nature. The normal distribution is defined by its bell-shaped curve, and the two most important characteristics of the normal distribution are the mean (μ) and the standard deviation (σ). The mean is the average of the distribution and the standard deviation is the "width" of the distribution around the mean. There are many phenomena in real life that obey Gaussian distribution, such as age, income, height, weight, etc. Most of them are in the middle level, and the proportion of especially few and especially many will be lower.

In Naive Bayes classification, the Gaussian model is used to deal with continuous-type feature variables. When using this model, we assume that the features belong to the Gaussian distribution, and then calculate the feature means and standard deviations based on the training samples so that the prior probability of each attribute value under this feature can be obtained.

Gaussian Naive Bayes is a classification technique for machine learning based on probabilistic methods and Gaussian distributions. Gaussian Naive Bayes assumes that each feature has the independent ability to predict the output variable. The combination of predictions for all parameters is the final prediction, which returns the probability that the dependent variable will be classified into each group, with the final classification being assigned to the group (class) with the higher probability.

Bayesian formula:
To make it easier to understand, instead of using event A and event B, we use the common machine learning expressions x, y to represent.

$$P(y_i|x) = P(x|y_i)P(y_i)/P(x)$$

According to the full probability formula, expanding the denominator P(A), we get

$$P(y_i|x) = P(x|y_i)P(y_i) / \sum_{1}^{m} P(x|y_j)P(y_j)$$

That is to say, if the feature x is known and you want to solve for yi, you only need to know the prior probability P(yi), and the likelihood P(x|yi) to solve for the posterior probability P(yi|x). And for the same sample x, P(x) is a constant and can be left out of the calculation.

Gaussian distribution:
If x is a continuous variable, how to go about estimating the likelihood P(x|yi)? We can assume that x obeys a Gaussian distribution (normal distribution) conditional on yi. P(x|yi) can be calculated from the probability density function of the normal distribution with the following formula.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Gaussian Parsimonious Bayes:
If x is multidimensional data, then we can assume that P(x1|yi),P(x2|yi)... . P(xn|yi) correspond to events that are independent of each other, and these values are multiplied together to get P(x|yi), "independent of each other" is the simplicity of Gaussian Bayes.

5.2 Decision Tree
Decision tree is an algorithm that classifies and predicts new data by measuring historical data. Simply put, a decision tree algorithm analyzes historical data that has clear results and looks for features in the data. This is used as a basis for predicting the outcome of newly generated data.

The decision tree consists of three main parts: decision nodes, branches, and leaf nodes. The topmost decision node of the decision tree is the root decision node. Each branch has a new decision node. Below the decision nodes are the leaf nodes. Each decision node represents a data category or attribute to be classified, and each leaf node represents an outcome.

Decision trees are a non-parametric supervised learning method for classification and regression. Its goal is to create a model that learns from data features and turn them into simple decision rules for inference.

A decision tree is a tree structure that divides the data by making a series of decisions (choices), which is similar to making choices for a series of problems. The decision process in a decision tree starts at the root node, tests the corresponding feature attributes in the item to be classified, and selects the output branches according to their values up to the leaf nodes, taking the stored categories of the leaf nodes as the decision result.

The process of generating a complete decision tree is the process of selecting what attributes to use as nodes, then there will be three types of nodes in the construction process.

Pruning:
Pruning is the process of slimming down the decision tree. The goal of this step is to get good results without too much judgment. The reason for this is to prevent the phenomenon of "overfitting".

There are several pruning methods:
Pre-pruning: Pruning is performed at the time of decision tree construction. This is done by evaluating the nodes during the construction process. If dividing a node does not lead to an improvement in accuracy in the validation set, then there is no point in dividing the node and the current node is treated as a leaf node and is not divided.
Post-pruning: Pruning is performed after the decision tree is generated. It usually starts from the leaf node of the decision tree and evaluates each node upwards layer by layer. If pruning this node subtree makes little difference in classification accuracy from keeping this node subtree, or if pruning this node subtree brings an improvement in accuracy in the validation set, then the node subtree can be pruned. This is done by replacing the node with a leaf node of this node subtree.

Classification decision tree based on C4.5 algorithm:
C4.5 is another classification decision tree algorithm based on the improved ID3 algorithm. C4.5 is an algorithm that inherits the advantages of the ID3 algorithm and the improved algorithm produces classification rules that are easy to understand and have high accuracy. At the same time, the algorithm also has some disadvantages, such as low efficiency of the algorithm and the value is suitable for data sets that can reside in memory. Based on the ID3 algorithm, the following improvements are being made to the C4.5 algorithm:

- The information gain rate is used to select attributes, which overcomes the deficiency of the ID3 algorithm in selecting attributes with a bias towards selecting attributes that take more values
- Pruning during the construction of the decision tree, which does not consider certain nodes with very good elements.
- Ability to complete the discrete processing of linked attributes.
- Capable of processing incomplete data.

5.3 Random Forest

Random forest model is a classifier that contain multiple decision trees. A random forest, as the name suggests, is a forest built in a random way, which consists of many decision trees that are not related to each other.

Random forest is an algorithm that integrates multiple trees through the idea of integrated learning, whose basic unit is the decision tree, and whose essence belongs to a branch of machine learning - Ensemble Learning. Ensemble learning is a machine learning method that uses a series of learners to learn, and integrates each learning method by a specific rule to obtain better learning results than a single learner. Integration learning solves a single prediction problem by building several models and combining them. It works mainly by generating multiple classifiers or models that each learn and make predictions independently.

Random forests are composed of multiple decision trees. For each tree, they use a training set that is sampled from the total training set using a put-back approach. For

each node of the tree, the features are randomly drawn from all the features in a proportional, non-replay back manner.

The random forest algorithm includes a repeated self-sampling process of the input data, known as bootstrap sampling. In this way, about one-third of the dataset will not be used for training the model but for testing, such data is called out of bag samples and the error estimated by these samples is called out of bag error.

It is shown that this out of bag method has the same degree of accuracy as the estimation method where the size of the test set is the same as the training set, so we do not need to set up the test set additionally in the random forest.

Advantages of Random Forest:
The random forest algorithm can solve both types of problems, classification and regression, and performs fairly well in estimating both.

Random Forest is excellent for high-dimensional data sets; it can handle thousands of input variables and identify the most important ones, and is therefore considered to be a good method for dimensionality reduction. In addition, the model is able to output the importance level of the variables, which is a very convenient feature.

Random forest is a very effective method when estimating missing data. Even if there is a large amount of missing data, random forest can maintain high accuracy.

When there is a class imbalance problem, random forests can provide an effective way to balance the error in the dataset.

Disadvantages of Random Forest:
Random forest does not perform as well as it does in classification when solving regression problems, because it does not give a continuous output. When performing regression, random forests are not able to make predictions beyond the training set data, which can lead to overfitting when modeling certain data with specific noise.

For many statistical modelers, random forest is like a black box - you have little control over the inner workings of the model, and have to experiment between different parameters and random seeds.

5.4 Logistic Regression

Logistic regression is one of the classification algorithms that use historical data to predict the probability of future outcomes. For example, we can set the probability of purchase as the dependent variable and the user's characteristics such as gender, age, registration time, etc. as the independent variables. The probability of purchase is predicted based on the characteristic attributes.

Logistic regression and multiple linear regression actually have a lot in common, the biggest difference is that their dependent variables are different, but the rest is basically the same. Because of this, the two regressions can be grouped into the same family of generalized linear models. The form of the models in this family is basically similar, the difference is that the dependent variable is different: if it is continuous, it is multiple linear regression; if it is binomial distribution, it is logistic regression; if it is Poisson

distribution, it is Poisson regression; if it is negative binomial distribution, it is negative binomial regression.

The dependent variable of logistic regression can be either dichotomous or multi-categorical, but dichotomous is more commonly used and easier to interpret. So in practice, the most commonly used logistic regression is the dichotomous logistic regression, and the independent variable can be either continuous or categorical.

6. Implementations of Classification Algorithms

The dataset was split into three parts: train set, validation set and test set. The train set was used for model building, the validation set was used to test the model performance during cross validation. And by comparing the result of each model, we could then select the best model for our solution. And the test set is used as unseen data points to test the performance of the best model.

As usual, we imported various classification algorithms from scikit-learn for implementation:

```
"""
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
"""
```

For Naive Bayes, we use the Gaussian Naive Bayes algorithm.

For Decision Tree, we use the hyperparameters of "max_depth", "min_samples_split" and "min_samples_leaf" to keep the model from overfitting.

For Random Forest, as it's composed of multiple decision trees, we use the same hyperparameters as the decision tree algorithm.

For Logistic Regression, we use the L2 penalty to penalize the insignificant predictors.

7. Comparison of Classification Results

Since we applied the ten-fold cross validation, each model has 10 results on ten trials. Later, we calculated the average values of these 10 results. Below is a table summarizing all the results:

	Metrics	Naive Bayes	Decision Tree	Random Forest	Logistic Regression
Trial 0	with intention precision	0.90	0.93	0.96	0.92
	with intention recall	0.44	0.91	0.89	0.92
	with intention f1-score	0.59	0.92	0.93	0.92
	precision macro average	0.55	0.74	0.76	0.73
	recall macro	0.59	0.76	0.84	0.73

	average				
	f1-score macro average	0.45	0.75	0.79	0.73
	accuracy	0.49	0.87	0.88	0.86
Trial 1	with intention precision	0.88	0.93	0.96	0.92
	with intention recall	0.57	0.90	0.90	0.92
	with intention f1-score	0.69	0.91	0.93	0.92
	precision macro average	0.54	0.74	0.76	0.74
	recall macro average	0.58	0.77	0.83	0.73
	f1-score macro average	0.50	0.75	0.79	0.73
	accuracy	0.58	0.86	0.88	0.86
Trial 2	with intention precision	0.87	0.93	0.95	0.92
	with intention recall	0.52	0.88	0.90	0.91
	with intention f1-score	0.65	0.91	0.93	0.92
	precision macro average	0.55	0.72	0.78	0.73
	recall macro average	0.58	0.76	0.84	0.74
	f1-score macro average	0.49	0.74	0.80	0.74
	accuracy	0.54	0.85	0.88	0.86
Trial 3	with intention precision	0.88	0.92	0.95	0.91
	with intention recall	0.52	0.90	0.90	0.92
	with intention f1-score	0.65	0.91	0.92	0.91
	precision macro average	0.54	0.72	0.75	0.74
	recall macro average	0.58	0.75	0.82	0.73
	f1-score macro average	0.48	0.73	0.78	0.73
	accuracy	0.54	0.85	0.87	0.85
Trial 4	with intention precision	0.88	0.92	0.95	0.92
	with intention recall	0.54	0.91	0.91	0.92
	with intention f1-score	0.67	0.92	0.93	0.92

	precision macro average	0.53	0.75	0.78	0.74
	recall macro average	0.57	0.76	0.83	0.74
	f1-score macro average	0.48	0.76	0.80	0.74
	accuracy	0.55	0.86	0.89	0.86
Trial 5	with intention precision	0.91	0.92	0.96	0.92
	with intention recall	0.51	0.90	0.90	0.91
	with intention f1-score	0.66	0.91	0.93	0.91
	precision macro average	0.55	0.72	0.76	0.71
	recall macro average	0.61	0.73	0.84	0.72
	f1-score macro average	0.48	0.72	0.79	0.71
	accuracy	0.54	0.85	0.89	0.85
Trial 6	with intention precision	0.89	0.92	0.96	0.92
	with intention recall	0.47	0.90	0.90	0.92
	with intention f1-score	0.62	0.91	0.93	0.92
	precision macro average	0.54	0.72	0.76	0.72
	recall macro average	0.58	0.75	0.83	0.73
	f1-score macro average	0.45	0.73	0.79	0.73
	accuracy	0.50	0.85	0.88	0.87
Trial 7	with intention precision	0.89	0.93	0.96	0.92
	with intention recall	0.49	0.90	0.89	0.92
	with intention f1-score	0.64	0.92	0.93	0.92
	precision macro average	0.55	0.72	0.78	0.71
	recall macro average	0.59	0.75	0.85	0.71
	f1-score macro average	0.48	0.74	0.80	0.71
	accuracy	0.53	0.86	0.88	0.86
Trial 8	with intention precision	0.88	0.93	0.95	0.92
	with intention	0.54	0.91	0.89	0.92

	recall				
	with intention f1-score	0.67	0.92	0.92	0.92
	precision macro average	0.54	0.72	0.75	0.74
	recall macro average	0.57	0.74	0.82	0.74
	f1-score macro average	0.48	0.73	0.78	0.74
	accuracy	0.55	0.86	0.87	0.86
Trial 9	with intention precision	0.89	0.92	0.96	0.91
	with intention recall	0.52	0.89	0.89	0.92
	with intention f1-score	0.66	0.91	0.92	0.91
	precision macro average	0.53	0.72	0.75	0.72
	recall macro average	0.57	0.75	0.84	0.71
	f1-score macro average	0.47	0.73	0.79	0.71
	accuracy	0.54	0.85	0.87	0.85
Average	with intention precision	0.89	0.92	0.96	0.92
	with intention recall	0.51	0.90	0.90	0.92
	with intention f1-score	0.65	0.91	0.93	0.92
	precision macro average	0.54	0.73	0.76	0.73
	recall macro average	0.58	0.75	0.84	0.73
	f1-score macro average	0.48	0.74	0.79	0.73
	accuracy	0.53	0.86	0.88	0.86

Since our goal is to detect those users who have purchasing intention, we focused on the metrics such as with intention precision, with intention recall and with intention f1-score.

We will use the result of random forest as illustrations:

With intention precision 0.96 means that out of all the predicted users with purchasing intention, 96% of the predictions are correct.

With intention recall 0.90 means that out of all the users with purchasing intention, 90% of them are detected.

With intention f1-score 0.93 is the harmonic mean of with intention precision and with intention recall.

In our case, we would like to detect as many positive users (users with purchasing intention) as possible. We don't really care if those predictions are wrong, because it doesn't hurt if we send ads to users who have no purchasing intention. That means that our key metric should be with intention recall.

Among the 4 classification models, 3 of them performed quite well - Decision Tree, Random Forest and Logistic Regression, with only slight differences between them.

And in the last step, we tested those three models on the test set, and they performed almost as well as on the validation sets. The results are shown below:

	Metrics	**Decision Tree**	**Random Forest**	**Logistic Regression**
Test set	with intention precision	0.92	0.95	0.90
	with intention recall	0.90	0.88	0.92
	with intention f1-score	0.91	0.92	0.91
	precision macro average	0.74	0.76	0.71
	recall macro average	0.76	0.83	0.69
	f1-score macro average	0.75	0.79	0.70
	accuracy	0.86	0.87	0.84

In summary, we would recommend using logistic regression for small to medium sized dataset and random forest for large dataset. And they are both very fast for real-time data analysis.

8. Recommendations

In the above use case, we identified those users who have purchasing intent using online browsing data, such as the number of different pages they viewed, the number of seconds they spent on those pages, exit rate, region, etc. If the company could acquire more information on their users, they could further increase the accuracy of the classification models, such information includes how much they earn per year, the number of their family members, how often they make the purchase, etc. And we could offer more customized content to each specific user. As the next step, we could also find out the characteristics of the users who have purchasing intention and finally place an order, such as the order time of the day, gender, region, etc. And we could then use that information for precision marketing.

Similarly, we could use similar techniques in other business sectors, such as call centres. If we want to use the classification method to analyse the characteristics of the customers who have called in, we need a detailed customer consultation dataset, including: inquiry time (month, week, time period, special day), inquiry Product, region,

customer gender, negative comments, whether to place an order, etc.

Because the asymmetry of information exchange between online shopping merchants and consumers makes it impossible for consumers to directly try the products by themselves, and they can only view the performance of the products through other channels. As a special form of online word-of-mouth, online reviews are an important channel for consumers to measure the quality of goods. Due to the anonymity of Internet information, many consumers are willing to post their real shopping experiences on the platform. A large number of objective and true effective information can help consumers comprehensively compare the advantages and disadvantages of products, and the referencing value for consumers to make purchase decisions is particularly significant. That is to say, consumers are more favourable to well-known enterprises and goods. This means that call centres that handle customer inquiries and complaints need to communicate as much as possible to get consumers to post positive reviews of the shopping experience in order to improve overall customer satisfaction.

Customer service is a series of behaviours designed to increase customer satisfaction, that is, to satisfy customers through warm services. E-commerce customer service covers all aspects of pre-sale, in-sale and after-sale. Unlike traditional industries, e-commerce customer service is mostly performed without direct face-to-face contact with customers, and the difficulty and complexity of service is greater than that of traditional industries. And customer service directly determines the development of the online store. Therefore, we should pay more attention to customer service. Do a good job in customer service to improve customer satisfaction with the store, improve customer experience, and attract customers. Good customer service can effectively reduce the loss of customers. If you don’t like this one, you can look at other styles or similar styles. There is always one that can retain your customers, as long as your customer service is professional and enthusiastic; The second purchase is the most concerned and the greatest value of many stores. For online stores, the second consumption or repeated consumption will reduce the promotion cost and greatly improve our profit margin.

The ultimate purpose of classifying data is to achieve precise marketing of online sales. The charm of precision marketing lies in precision. When we are doing marketing to acquire customers, our biggest expectation is to make the marketing content directly in front of target users, plus a good content display to make them interested, and finally they would be eager to buy our products. The biggest factor that determines the customer acquisition results lies in the funnel diagram. If 1,000 visitors come to your website or APP, and 900 of them showed interest, then the number of customers we get in the end must be high. of. But if there are also 1000 visitors, with only 100 people are your potential customers, then you will find that no matter how good your product is, only small number of users would make the purchase. If we target the wrong people for marketing, it is like looking for a needle in a haystack, and a large amount of marketing expenses will be wasted.

To sum up, we recommend that the customer consultation data of the call centre be aggregated and pre-processed first, then use classification methods to classify customers, and find out the characteristics of different categories of customers. At the same time, carry out intelligent marketing for different types of customers. We perform feature portraits by analysing the characteristics of users who have completed

behaviours in the past. For example, a user who purchased a product has seen a company introduction video and two cases in the past, then we can determine that if the user has watched the video and the case, they will be more willing to go. After completing the purchase, we will send mass messages to users who have had this behaviour in the past period of time. Such mass messaging will be relatively more accurate.

9. Ethics

As the role of data science in the business world grows, so do the ethical and legal issues associated with it. Then companies that want to use data to support their business should be aware of the ethical and legal risks associated with collecting user data to avoid the resulting damage to the company, both monetary and reputational. For example, it is important to obtain consent from users to collect, store and use their data. And the company should establish a complete firewall system and technology for network security, and pay attention to the latest network security issues and keep the security technology updated to protect users' data from being stolen or leaked. In addition, in the process of collecting user data, attention should be paid to what data is sensitive, even if the user is asked for consent beforehand, which sometimes makes the user feel uncomfortable and thus leads to adverse effects on the company's operation and reputation, such as collecting data on the user's race, gender, etc.

In summary, data protection shall be designed according to the GDPR regulation and whenever and however you collect personal data, you have both ethical and legal obligations to ensure that participants' information is properly protected.

10. Conclusion

As shown in this use case, we trained several classification models to predict the user purchasing intent, and classify them into two classes - with purchasing intent & without purchasing intent. And we are able to identify about 90% of the users who have purchasing intent among all the users using the best model, and our prediction accuracy is about 90% as well, which means that out of all of the users we predicted to be having purchasing intent, 90% of them actually have purchasing intent. Our model is a success on this use case dataset, and thus can be proved to be able to add values to the company which is planning to implement data analytics and machine learning to improve their businesses. Furthermore, as stated in our recommendations, they could find out the common characteristics in these users who have purchasing intent and target them for precision marketing. They could also implement similar technologies in other business sectors, such as customer service / call centres.

References

Sakar, C.O., Polat, S.O., Katircioglu, M. and Kastro, Y., 2019. Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. Neural Computing and Applications, 31(10), pp.6893-6908.

Kuyumdzhieva, A., 2019. General Data Protection Regulation and Horizon 2020 Ethics Review Process: Ethics Compliance under GDPR. Bioethica, 5(1), p.6.