PASREFT(Prediction, Ad-Optimization, Segmentation and Recommendation Engine for Tech_Novan)

22.12.2021

Nishant Wadhwani
Machine Learning Engineer

Overview

Data science and Machine Learning have seen applications across all industries including e-commerce. According to a report by a leading newspaper, India is the fastest growing online retail among the top global economies. Having a growth rate of more than 50%, more e-commerce websites have surfaced than in the past. Currently, most of the E-commerce specific advancements revolve around the applications of Data science and Machine Learning to come up with innovative solutions to complex problems.

With roughly 2.5 quintillion bytes of data generated each day, it's become crucial for these players to use it in a way to keep the customers happy and satisfied. This is where the importance of data science projects comes into the picture, where they are using it in areas such as customer segmentation, recommender system, ad optimization, and more. For instance, eBay is investing ample funds into data science and personalizing shopbots for enhancing customers' experiences. Whereas Myntra is using data analytics and AI through its bot Artie, which is a smart bot for gathering consumer insights, and more.

Goals

Let's suppose we just got some contract work with an E-commerce company named Tech_novan based in Bangalore City, an online hub for electronic gadgets and computer products. The website has a wide range of products ranging from laptops, tablets, Smartphones, Digital Cameras to Gaming hardware, monitors to printers, routers, and speakers. Shoppers can browse and buy a wide variety of electronics on the extensive but easy-to-navigate e-commerce site.

The navigation and categorization are well defined, giving you suggestions for keywords as you type. In addition to this, they also have in-store sessions related to electronic gadgets or devices' functionality and provide feedback related to a particular model in these advice sessions. Customers come into the store, have sessions/meetings with a personal techie, then they can go home and order either on a mobile app or website for the device/gadget they want. Now, as a Data Scientist/ML Engineer, we are going to build a system called PASREFT(Prediction, Ad-Optimization, Segmentation and Recommendation Engine for Tech_Novan) in which we have to perform the following seven tasks for this company to maintain their overall revenue and marketing/advertisement cost, improvising the system to get the best results.

Prediction System to increase revenue using Regression:

Specifications

The company is trying to decide whether to focus its efforts on its mobile app experience or its website. To solve this issue, we will keep a record of the average time spent on the app in minutes, average time spent on the website in minutes, length of membership of the peculiar customer to determine and predict the yearly amount spent by the customer by building various machine learning models.

We will first do exploratory data analysis to get an idea about the data we have, to understand the relationship between different features, and also based upon the plots we will try to anticipate what can be the final answer for the problem so that even the board members in the organization can understand it better. After that, we will do data pre-processing that includes data cleaning, splitting of data, feature scaling, and encoding the categorical data. After that, we will start with the building of machine learning algorithms. For this part, since our data has only four features and is important for getting the dependent variable, feature selection is not necessary for this task.

We will use eleven types of regression techniques which include Multiple Linear Regression, Polynomial Regression, Decision Tree Regression, Random Forest Regression, Support Vector Regression, AdaBoost Regression, Gradient Boosting Regression, LightBoost Regression XGBoost Regression, Catboost Regression, and a Deep Learning Neural Network with two hidden layers. To improve the model performance, we will also apply model selection techniques like K-Fold Cross-Validation and Hyperparameter Tuning using Randomized Search and Grid Search. In the end, we will also compare these models by checking the error functions like MAE, MSE, RMSE, and R2_Score(since the data is linear) to get the best model. This dataset will contain details of 500 customers who use these online platforms at regular intervals of time.

Features Used:-

Avg. Session Length: Average session of in-store style advice sessions.

Time on App: Average time spent on App in minutes

Time on Website: Average time spent on Website in minutes

Length of Membership: How many years the customer has been a member.

Email-id of the Customer

Address of the Customer

Yearly Amount Spent by the Customer in dollars:- Dependent Variable

Milestones

Results:-

1. Multiple Linear Regression:-

MAE: 7.645674798915295

MSE: 92.89010304498548

RMSE: 9.637951185028149

AdjR2_Score: 98.60 %

R2_Score: 98.28 %

Standard Deviation: 0.23 %

2. Polynomial Regression:-

MAE: 9.821412017434954

MSE: 168.1146222773757

RMSE: 12.965902293221852

AdjR2_Score: 0.9748088085532581

R2_Score: 98.28 %

Standard Deviation: 0.23 %

3. Decision Tree Regression:-

MAE: 22.302801428455044

MSE: 892.9177245951145

RMSE: 29.881728942534675

AdjR2_Score: 0.8638033002783084

R2_Score: 85.47 %

Standard Deviation: 4.47 %

4. Random Forest Regression:-

MAE: 16.792435420526388

MSE: 588.0010281930992

RMSE: 24.24873250693939

AdjR2_Score: 0.9118907791584572

R2_Scores: 91.40 %

Standard Deviation: 2.18 %

5. Gradient Boosting Regression:-

MAE: 11.995227880605066

MSE: 300.38339692348706

RMSE: 17.331572257688773

AdjR2_Score: 0.9549889442574704

R2_score: 95.78 %

Standard Deviation: 1.23 %

6. AdaBoost Regression

MAE: 23.401292929729806

MSE: 1057.8274956580312

RMSE: 32.52426010930965

AdjR2_Score: 0.8414894669255895

R2_score: 87.26 %

Standard Deviation: 2.46 %

7. LightGBM Regression

MAE: 17.07568894454065

MSE: 597.5295569718282

RMSE: 24.444417705722266

AdjR2_Score: 0.910462973412879

R2_score: 93.68 %

Standard Deviation: 1.89 %

8. Support Vector Regression:-

MAE: 7.684301593806611

MSE: 93.41165186472095

RMSE: 9.664970349914217

AdjR2_Score:0.9861149009981192

R2_Score: 98.28 %

Standard Deviation: 0.23 %

9. XGBoost Regression:-

MAE: 12.416217399641823

MSE: 300.99728221487806

RMSE: 17.349273247455585

AdjR2_Score: 0.9530468237185467

R2_Score: 96.55 %

Standard Deviation: 0.98 %

10. CatBoost Regression:-

MAE: 11.80193070072924

MSE: 290.54804106706683

RMSE: 17.04546981068773

AdjR2_Score: 0.9564627266144017

R2_Score: 96.29 %

Standard Deviation: 1.07 %

11. Deep-Learning Neural Network:-

MAE: 8.977470664848756

MSE: 145.9051320075004

RMSE: 12.079119670220194

R2_Score: 97.25%

Standard Deviation: 0.994%

So, as we can see from the following results Multiple Linear Regression and Support Vector Regression have the best performance if we look at the final values, and it is because the data is somehow linearly separable as in this case relationship between the yearly amount spent by the customer and length of membership of a particular customer is linearly separable as well as the relationship between the yearly amount spent by the customer and time spent on the app by the customer has a linear relationship.

And the conclusions we derived are:-

Holding all other features fixed, a 1 unit increase in Avg. Session Length is associated with an increase of 25.98 total dollars spent by the customer.

Holding all other features fixed, a 1 unit increase in Time on App is associated with an increase of 38.59 total dollars spent by the customer.

Holding all other features fixed, a 1 unit increase in Time on Website is associated with an increase of 0.19 total dollars spent by the customer.

Holding all other features fixed, a 1 unit increase in Length of Membership is associated with an increase of 61.27 total dollars spent by the customer.

For this particular scenario, a quick question:- Do you think the company should focus more on its mobile app or its website?

This is tricky, there are two ways to think about this: Develop the Website to catch up to the performance of the mobile app, or develop the app more since that is what is working better. This sort of answer depends on the other factors going on at the company, you would probably want to explore the relationship between Length of Membership and the App or the Website before concluding!

Ad-Optimization using Reinforcement Learning:

Specifications

In this particular task, we're going to optimize the online advertising by finding the best ad among different ad designs, the best ad that will converge the maximum customers to click on the ad and potentially buy the company's product.

The advertising team prepared ten different ads with ten different designs. For example one of the ads the customers will see the smartphone commercial done by famous cricketer Virat Kohli, who is doing heavy work-out on the field and recording/monitoring his performance on some app or comparing his metabolism with battery-life, on the other ad customers see the smartwatch commercial done by renowned actor Brad Pitt during some action scene while chasing criminals, and on some other ad say some tourists models are using a digital camera which is taking HD photographs of an SUV in a charming city like the south of France or Italy, etc. Now, the advertising team is wondering which ad will converge the most, which ad will attract the most people to click the ad and then potentially buy their products.

So we have these ten different ads and what we're going to do is the process of online learning, which is to show these ads to different users online. Once they connect to a certain website or a search engine, it can be ads that appear at the top of a page when they do any type of research on Google. We will show one of these ads each time the user connects to the Web page and we're going to record the result whether this user clicked yes or no on the ad.

We will be using reinforcement learning for this task that includes Upper Confidence Bound and Thompson Sampling for getting the final result. So, we will select an ad to show to this user and then the user will decide to click yes or no on the ad. If the user clicks on the ad we will record it as one whereas if the user doesn't click on the ad we will record it as 0. And then a new user connects to the Web page and the algorithm selects an ad to show this new user. Our dataset contains 10000 users. We need to figure out in a minimum number of rounds which ad converts to the most meaning which is the best ad to which the users are most attracted.

Milestones

Ad number five was the ad that was selected the most and was the ad with the highest click-through rate. In terms of our business case study, it corresponds to the ad that is the most attractive ad that has the most fanbase associated, that will sell the most to the user as future customers.

The UCB algorithm did a great job but here is a constraint that is to identify this ad as soon as possible which is in a minimum number of rounds. We have to observe how many rounds are required for the UCB algorithm to be able to identify this ad with the highest clicks. We will implement this by tweaking the number of customers as a parameter. With 5000 rounds that is with 5000 users, the UCB can identify the ad with the highest CTR. If we replace 5000 here with 1000 and still it can identify the ad with the highest CTR which is still ad 5. Now, we're going to replace that 1000 here with 500. And 500 rounds is not enough for the UCB algorithm to identify the best ad, the UCB identified the best ad as an ad of index 7 so 500 is not enough.

Now, if we use the Thompson Sampling algorithm with the same data set then it will give the correct result even with 500 users i.e. it was able to find the ad number 5 with the highest CTR even in 500 rounds. And therefore this technique is more powerful and efficient than UCB in most situations.

Prediction System for Best Ad using Classification:

Specifications

Now, as we got our best ad, that is ad no.5, we are going to test this ad for our new set of customers. We will be working with an advertising data set that contains 1000 new users and indicating whether or not a particular internet user clicked on an advertisement on any website or app. These advertisements will be related to our company's product that can be any electronic gadget or device because there's a high chance that these new customers will click on this ad since it has got maximum CTR when tested on our old customers. And if a person clicks on this best ad, there is a possibility that he/she will purchase our product.

We will try to create a model that will predict whether or not they will click on an ad based on the features of that user. We'll work with the Ecommerce Customers CSV file from the company. It has Customer info, such as Age, Daily Internet Usage, and their respective countries. This data set contains the following features:

Features Used:-

Age: customer age in years

Area Income: Avg. Income of geographical area of consumer

Daily Internet Usage: Avg. minutes a day consumer is on the internet

Ad Topic Line: Headline of the advertisement

City: City of consumer

Male: Whether or not consumer was male

Country: Country of consumer

Timestamp: Time at which consumer clicked on Ad or closed window

Clicked on Ad: 0 or 1 indicated clicking on Ad

We will first do exploratory data analysis to get an idea about the data we have, to understand the relationship between different features, and also based upon the plots we will try to anticipate what can be the final answer for the problem so that even the board members in the organization can understand it better. After that, we will do data pre-processing that includes data cleaning, splitting of data, feature scaling, and encoding the categorical data. After that, we will start with the building of machine learning algorithms. For this part, since our data has only five features and that too is important for getting the dependent variable, feature selection is not necessary for this task.

We will use twelve types of classification techniques which include Logistic Regression, KNearestNeighbour Classification, Naive Bayes Classification, Decision Tree Classification, Random Forest Classification, Support Vector Machine Classification, XGBoost Classification, Gradient Boosting Classification, AdaBoost Classification, LightGBM Classification, Catboost Classification, and a Deep Learning Neural Network with two hidden layers. To improve the model performance, we will also apply model selection techniques like K-Fold Cross-Validation and Hyperparameter Tuning using Randomized Search and Grid Search. In the end, we will also compare these models by comparing their accuracies to get the best model. This dataset contains details of 1000 internet users who use these online platforms at regular intervals of time.

Milestones

Results Obtained:-

1. Logistic Regression:-

Accuracy: 96.53 %

Standard Deviation: 1.48 %

```
          precision   recall  f1-score   support


0         0.97      0.99      0.98       136

1         0.99      0.96      0.98       114
```

2. KNearestNeighbour Classification:-

Accuracy: 96.27 %

Standard Deviation: 1.55 %

```
  precision   recall  f1-score   support


0      0.95      1.00      0.97       136

1      1.00      0.94      0.97       114
```

3. SVM Classification:-

Accuracy: 96.80 %

Standard Deviation: 1.36 %

```
  precision   recall  f1-score   support


0      0.97      0.99      0.98       136

1      0.99      0.96      0.98       114
```

4. Naive Bayes Classification:-

```
Accuracy: 96.27 %
Standard Deviation: 1.31 %
  precision    recall  f1-score   support

0       0.97      0.97      0.97       136
1       0.96      0.96      0.96       114
```

5. Decision Trees Classification:-

```
Accuracy: 95.20 %
Standard Deviation: 1.74 %
  precision    recall  f1-score   support

0       0.92      0.95      0.93       136
1       0.94      0.90      0.92       114
```

6. Random Forest Classification:-

```
Accuracy: 96.00 %
Standard Deviation: 2.23 %
  precision    recall  f1-score   support

0       0.96      0.96      0.96       136
1       0.96      0.95      0.95       114
```

7. AdaBoost Classification:-

```
Accuracy: 94.80 %
Standard Deviation: 2.10 %
  precision   recall  f1-score   support

0      0.97      0.97      0.97      136
1      0.96      0.96      0.96      114
```

8. Gradient Boosting Classification:-

```
Accuracy: 95.47 %
Standard Deviation: 1.48 %
  precision   recall  f1-score   support

0      0.96      0.97      0.96      136
1      0.96      0.95      0.96      114
```

9. LightGBM Classification:-

```
Accuracy: 95.20 %
Standard Deviation: 1.60 %
  precision   recall  f1-score   support

0      0.96      0.97      0.96      136
1      0.96      0.95      0.96      114
```

10. XGBoost Classification:-

```
Accuracy: 95.73 %
Standard Deviation: 2.05 %
  precision   recall  f1-score   support
```

0	0.98	0.97	0.97	136
1	0.97	0.97	0.97	114

11. CatBoost Classification:-

Accuracy: 95.87 %

Standard Deviation: 1.73 %

	precision	recall	f1-score	support
0	0.96	0.98	0.97	136
1	0.97	0.96	0.96	114

12. Deep Learning Neural Network:-

Accuracy: 97.2%

	precision	recall	f1-score	support
0	0.96	0.99	0.98	136
1	0.99	0.96	0.97	114

So, as we can see Support Vector Machine and ANN has correctly predicted the result with the highest accuracy i.e. 96.80 and 97.2 as well as good precision, recall value which is using RBF as a kernel and NN with two hidden layers which represent that the data here is non-linear in nature. Also, we got the magnificent result that out of 520 women users, around 260 have clicked our ad to see the specific gadget and out of 480 men, around 230 have clicked our ad which shows in total around half of the new users found our ad engaging.

Prediction System for Purchasing Product using Classification:

Specifications

Now, after solving all these tasks, we are interested in the main problem that is whether the person is interested in buying the product or not i.e. whether the particular customer after attending all the brainstorming sessions and clicking on the ad that has the highest CTR, has purchased any of the company's product. So, using classification, we are going to predict whether the customer has purchased the product or not based on two features that are the age of the customer and his estimated salary in dollars per annum. So, it is quite clear that from our previous dataset, around 500 customers have clicked the best design ad, so we are taking the information that is age and estimated salary of 400 customers.

We will first do exploratory data analysis to get an idea about the data we have, to understand the relationship between different features, and also based upon the plots we will try to anticipate what can be the final answer for the problem so that even the board members in the organization can understand it better. After that, we will do data pre-processing that includes data cleaning, splitting of data, feature scaling, and encoding the categorical data. After that, we will start with the building of machine learning algorithms. For this part, since our data has only two features and that too is important for getting the dependent variable, feature selection is not necessary for this task.

We will use twelve types of classification techniques which include Logistic Regression, KNearestNeighbour Classification, Naive Bayes Classification, Decision Tree Classification, Random Forest Classification, Support Vector Machine Classification, XGBoost Classification, Gradient Boosting Classification, AdaBoost Classification, LightGBM Classification, Catboost Classification, and a Deep Learning Neural Network with two hidden layers. To improve the model performance, we will also apply model selection techniques like K-Fold Cross-Validation and Hyperparameter Tuning using Randomized Search and Grid Search. In the end, we will also compare these models by comparing their accuracies to get the best model. This dataset contains details of 400 users i.e their age and salaries.

Milestones

Results Obtained:-

1. Logistic Regression:-

Accuracy: 82.67 %

Standard Deviation: 9.52 %

```
  precision    recall  f1-score   support

0       0.89      0.96      0.92        68
1       0.89      0.75      0.81        32
```

2. KNearestNeighbour Classification:-

Accuracy: 91.00 %

Standard Deviation: 5.59 %

```
   precision    recall  f1-score   support

0       0.96      0.94      0.95        68
1       0.88      0.91      0.89        32
```

3. SVM Classification:-

Accuracy: 90.67 %

Standard Deviation: 6.11 %

```
  precision    recall  f1-score   support

0       0.96      0.94      0.95        68
1       0.88      0.91      0.89        32
```

4. Naive Bayes Classification:-

Accuracy: 87.67 %

Standard Deviation: 8.95 %

```
   precision    recall  f1-score   support

0       0.90      0.96      0.93       68
1       0.89      0.78      0.83       32
```

5. Decision Trees Classification:-

Accuracy: 90.67 %

Standard Deviation: 6.29 %

```
  precision    recall  f1-score   support

0       0.97      0.94      0.96       68
1       0.88      0.94      0.91       32
```

6. Random Forest Classification:-

Accuracy: 91.00 %

Standard Deviation: 6.33 %

```
  precision    recall  f1-score   support

0       0.97      0.94      0.96       68
1       0.88      0.94      0.91       32
```

7. XGBoost Classification:-

Accuracy: 88.00 %

Standard Deviation: 6.18 %

```
  precision   recall  f1-score   support

0      0.96      0.94      0.95       68
1      0.88      0.91      0.89       32
```

8.AdaBoost Classifier

Accuracy: 85.67 %

Standard Deviation: 7.16 %

```
  precision   recall  f1-score   support

0      0.93      0.94      0.93       68
1      0.87      0.84      0.86       32
```

9.Gradient Boosting Classifier

Accuracy: 88.67 %

Standard Deviation: 5.81 %

```
  precision   recall  f1-score   support

0      0.93      0.94      0.93       68
1      0.87      0.84      0.86       32
```

10.LightGBM Classifier

Accuracy: 90.00 %

Standard Deviation: 6.50 %

```
  precision    recall  f1-score   support


0      0.93      0.91      0.92       68

1      0.82      0.84      0.83       32
```

11. CatBoost Classification:-

Accuracy: 91.00 %

Standard Deviation: 5.97 %

```
  precision    recall  f1-score   support


0      0.96      0.94      0.95       68

1      0.88      0.91      0.89       32
```

12. Deep Learning Neural Network:-

Accuracy: 89.0

Standard Deviation: 5.972

```
   precision    recall  f1-score   support


0      0.94      0.91      0.93       68

1      0.82      0.88      0.85       32
```

So, as we can see KNN Classification has predicted with best average accuracy i.e. 91% with standard deviation of 5.57% (and also precision, recall values are also good) whether the customer will buy the product or not after clicking the ad with the best design or the ad with maximum click-through rates. It means our model is 91 percent efficient in predicting the correct number of users who have either purchased the product or not. Also, out of the 400 customers, around 140 customers have brought something for them either from the company's website or app or through visiting the store which is tremendously contributing to the company's profit.

Customer Segmentation using Clustering:

Specifications

The strategy team of Tech_Novan collected some data about their customers. Each row corresponds to a customer of the organization. And for each of these customers, the data analyst of this team gathered the following information that includes the customer's I.D., their gender, their age, their annual income, and then the spending score that can take values between 1 and 100. The spending score is a metric made by the company to measure how much each customer spends. And so they made this metric taking values from 1 to 100. The scale of the metric is such that the lower the score, the less the customer spends, and the higher the score the more the customer spends. And this is in a certain period, let's say in the past year.

So now the purpose of this task is the strategic team or analytics team wants to understand its customers. That is, they want to identify some patterns within their base of customers using some machine learning algorithms like the clustering technique in this case. But even though we don't know what specifically to predict, we still know that we want to identify some patterns and do customer segmentation.

How are we going to identify such patterns?

Well, we will do this by using K-means Clustering and Hierarchical Clustering, and more specifically what we will do is we will create a dependent variable that will take a finite number of values.

Milestones

We have these different classes of customers and so now we know some business strategies or business ideas like in marketing that would be to target the company's customers. The customers who belong to cluster three, are characterized by a high annual income with a high spending score, for these customers, we could truly target them the most when there are some new offers like new deals because indeed from these customers we'll get the highest chance to sell our products since they spend a lot. And besides, they have a high annual income, therefore, a high potential to buy a lot of stuff.

Cluster number one is the cluster of the customers who earn a low annual income and yet have a high spending score. For these customers who seem to earn a low annual income but yet seem to have a problem spending too much, maybe the organization would like to be responsible and protect these customers by not targeting them too much with new deals and new irresistible offers that after all maybe these customers don't need at the end of the day. They would limit perhaps the targeting to these customers. Cluster number four contains all the customers who have a low annual income and who spend very little on the company's website or app. Well, the board of the organization should not do anything with this cluster because we don't have to protect them. After all, they don't spend much. And besides, they have a low income, so we don't want to target them too much. Then the cluster number five corresponds to all the customers who earn a high annual income but still who don't spend much on purchasing the company's products.

This would be interesting to target because maybe we're missing out on a lot of customers who don't seem to show that much interest in the company's products here. So maybe for these classes, we could brainstorm on how to send them some better advertising that attracts them more and track them better into the company's website or for in-store mall sessions so that they can purchase more products and increase their spending score.

Now, the final cluster which is like an average cluster and contains customers who earn an average annual income and spend normally on the company's website as well as come sometimes for getting in-store advice sessions. And for this peculiar class, we cannot do much as we don't want to target the ones that have a low annual income. So we want to protect them at the same time.

So, here we identified some different clusters of customers and for each of them, we can deploy different marketing strategies or different business strategies which will boost customers in some of the clusters and will protect them in other clusters.

Recommender System using Collaborative Filtering and Association Rule Mining:

Specifications

E-commerce companies like Amazon, Flipkart, Snapdeal use different recommendation systems to provide suggestions to the customers. Amazon currently uses item-item collaborative filtering, which scales to massive datasets and produces a high-quality recommendation system in real-time. Tech_Novan board members have planned to make a recommender system that kind of conveys an information filtering system that seeks to recommend the products which the user is interested in.

In this modern world, we are overloaded with data and this data provides us with useful information. But the user cannot extract the information which interests them from this data. To help the user to find out information about the product, recommendation systems were developed.

The recommender system creates a similarity between the user and items and exploits the similarity between user/item to make recommendations.

What type of recommender system and how does the recommender system solve this problem?

1. It can help the user to find the right product.
2. It can increase user engagement. For example, there's 40% more click on google news due to recommendations.
3. It helps the item providers to deliver the items to the right user. On Amazon, 35 % of products get sold due to recommendations.
4. It helps to make the contents more personalized. On Netflix, most of the rented movies are from recommendations.

There are mainly 3 types of recommendations systems we will be implementing in this project:-

-> Popularity-based systems:- It works by recommending items viewed and purchased by most people and are rated high. It is not a personalized recommendation.

->Collaborative Filtering:- It is based on the assumption that people like things similar to other things they like, and things that are liked by other people with similar tastes. it is mainly of two types: a) User-User b) Item -Item

->Association rule mining:- Association rules capture the relationships between items based on their patterns of co-occurrence across transactions.

We will be taking around 20,000 entries from 7.8 million reviews spanning May 1996 - July 2014 considering the CPU and GPU specifics of this machine.

Dataset will contain these features:

1.userId	object
2.productId	object
3.Rating	float64
4.timestamp	int64

-->Firstly, We as data scientists would like to implement Popularity Based Recommendation Systems which will work with the trend for Tech_Novan. It uses the items which are in trend right now. For example, if any product is usually bought by every new user then there are chances that it may suggest that item to the user who just signed up.

The problem with the popularity-based recommendation system is that personalization is not available with this method i.e. even though you know the behavior of the user you cannot recommend items accordingly.

And we will be filtering the dataset containing users who have given 5 or more ratings and the products which got at least 5 ratings. After filtering the dataset and making a column of the total rating count for products, we will be recommending the 30 most used products with the top 30 total rating count as the criteria.

-->Secondly, we will implement Collaborative filtering (Item-Item recommendation) which is commonly used for recommender systems. These techniques aim to fill in the missing entries of a user-item association matrix. CF is based on the idea that the best recommendations come from people who have similar tastes. In other words, it uses historical item ratings of like-minded people to predict how someone would rate an item.

Collaborative Filtering is a technique or a method to predict a user's taste and find the items that a user might prefer based on information collected from various other users having similar tastes or preferences. It takes into consideration the basic fact that if person X and person Y have a certain reaction for some items then they might have the same opinion for other items too.

The two most popular forms of collaborative filtering are:

User-Based: Here, we look for the users who have rated various items in the same way and then find the rating of the missing item with the help of these users.

Item Based: Here, we explore the relationship between the pair of items (the user who bought Y, also bought Z). We find the missing rating with the help of the ratings given to the other items by the user.

Let’s talk about Item-Based Collaborative Filtering in detail. It was first invented and used by Amazon in 1998. Rather than matching the user to similar customers, item-to-item collaborative filtering matches each of the user’s purchased and rated items to similar items, then combines those similar items into a recommendation list. In a system where there are more users than items, item-based filtering is faster and more stable than user-based. It is effective because usually, the average rating received by an item doesn’t change as quickly as the average rating given by a user to different items. It’s also known to perform better than the user-based approach when the rating matrix is sparse.

Item to Item Similarity: The very first step is to build the model by finding similarity between all the item pairs. The similarity between item pairs can be found in different ways. One of the most common methods is to use cosine similarity.

Formula for Cosine Similarity:

Similarity(A, B)= A.B/(||A||*||B||)

Prediction Computation: The second stage involves executing a recommendation system. It uses the items (already rated by the user) that are most similar to the missing item to generate ratings. We hence try to generate predictions based on the ratings of similar products. We compute this using a formula which computes a rating for a particular item using a weighted sum of the ratings of the other similar products.

rating(U, I_i)= Summation_over_j{rating(U, I_j)*s_ij}/Summation_over_j{s_ij}

where s_ij refers to the similarity between ith and jth item

Example:

Let us consider one example. Given below is a set table that contains some items and the user who has rated those items. The rating is explicit and is on a scale of 1 to 5. Each entry in the table denotes the rating given by a ith user to a jth Item. In most cases the majority of cells are empty as a user rates only for a few items. Here, we have taken 4 users and 3 items. We need to find the missing ratings for the respective user.

User/Item	Item_1	Item_2	Item_3
User_1	2	–	3
User_2	5	2	–
User_3	3	3	1
User_4	–	2	2

Step 1: Finding similarities of all the item pairs.

Form the item pairs. For example in this example the item pairs are (Item_1, Item_2), (Item_1, Item_3), and (Item_2, Item_3). Select each item to pair one by one. After this, we find all the users who have rated for both the items in the item pair. Form a vector for each item and calculate the similarity between the two items using the cosine formula stated above.

Sim(Item1, Item2)

In the table, we can see only User_2 and User_3 have rated for both items 1 and 2.

Thus, let I1 be the vector for Item_1 and I2 be for Item_2. Then,

I1 = 5U2 + 3U3 and,

I2 = 2U2 + 3U3

Similarity(I1, I2) = {(5*2) + (3*3)}/{(sqrt(5^2 + 3^2) *(sqrt(2^2 + 3^2)} = 0.90

Sim(Item2, Item3)

In the table we can see only User_3 and User_4 have rated for both the items 1 and 2.

Thus, let I2 be the vector for Item_2 and I3 be for Item_3. Then,

I2 = 3U3 + 2U4 and,

I3 = 1U3 + 2U4

Similarity(I2, I3) = {(3*1) + (2*2)}{(sqrt(3^2 + 2^2) * (sqrt(1^2 + 2^2)} = 0.869

Sim(Item1, Item3)

In the table we can see only User_1 and User_3 have rated for both the items 1 and 2.

Thus, let I1 be the vector for Item_1 and I3 be for Item_3. Then,

I1 = 2U1 + 3U3 and,

I3 = 3U1 + 1U3

Similarity(I1, I3) = {(2*3) + (3*1)}/{(sqrt(2^2 + 3^2) * (sqrt(3^2 + 1^2)} = 0.789

Step 2: Generating the missing ratings in the table

Now, in this step we calculate the ratings that are missing in the table.

Rating of Item_2 for User_1

r(U_1, I_2)= {r(U_1, I_1)*s_{I_1,I_2} + r(U_1, I_3)*s_{I_3,I_2} / {s_{I_1, I_2} + s_{I_3,I_2}}= {(2*0.9) + (3*0.869)/{(0.9 + 0.869)} = 2.49

Rating of Item_3 for User_2

r(U_2, I_3)= {r(U_2, I_1)*s_{I_1,I_3} + r(U_2, I_2)*s_{I_2,I_3}/ {s_{I_1,I_3} + s_{I_2,I_3}}= {(5*0.789) + (2*0.869)}{(0.789 + 0.869)} = 3.43

Rating of Item_1 for User_4

r(U_4, I_1)= {r(U_4, I_2)*s_{I_1,I_2} + r(U_4, I_3)*s_{I_1,I_3}/ {s_{I_1,I_2} + s_{I_1,I_3}}= {(2*0.9) + (2*0.789)}{(0.9 + 0.789)} = 2.0

To implement an item based collaborative filtering, KNN is a perfect go-to model and also a very good baseline for recommender system development. But what is the KNN? KNN is a non-parametric, lazy learning method. It uses a database in which the data points are separated into several clusters to make inference for new samples.

KNN does not make any assumptions on the underlying data distribution but it relies on item feature similarity. When KNN makes inference about a product, KNN will calculate the "distance" between the target product and every other product in its database, then it ranks its distances and returns the top K nearest neighbor products as the most similar product recommendations.

For the Collaborative filtering(Item-Item recommendation), we will be using the k-Nearest Neighbors (kNN) algorithm. kNN is a machine learning algorithm to find clusters of similar users based on common product ratings, and make predictions using the average rating of top-k nearest neighbors. For example, we first present ratings in a matrix with the matrix having one row for each item and one column for each user.

When we work with kNN — type recommender algorithms, there are 2 hyperparameters we can tune: the k parameter (yes, same k as in the name of the model type), and the similarity option. The k parameter is fairly straightforward, and analogous to how it works in general k-nearest neighbors models: it is the upper limit of similar items we want the algorithm to consider. For example, if the user rated 20 games, but we set k to 10, when we estimate a rating to a new game, only those 10 games out of 20 that are the closest to the new game will be considered. You can also set min_k, if a user does not have enough ratings, the global average will be used for estimations. As a default, it's 1.

We mentioned items being close to each other in the previous paragraph, but how do we determine that distance? It is the second hyperparameter, the similarity option, that defines the way to calculate it.

All the similarity functions will return a number between 0 and 1 to a specific (i, j) item pair. And that number corresponds to the distance between the two items, the nearer the value to 0, the nearer will be the distance between items and maximum will be the correlation between them.

Now our training data has a very high dimensionality. kNN's performance will suffer from curse of dimensionality if it uses "euclidean distance" in its objective function. Euclidean distance is unhelpful in high dimensions because all vectors are almost equidistant to the search query vector (target movie's features). Instead, we will use cosine similarity for nearest neighbor search.

Item-based: For an item I, with a set of similar items determined based on rating vectors consisting of received user ratings, the rating by a user U, who hasn't rated it, is found by picking out N items from the similarity list that have been rated by U and calculating the rating based on these N ratings.

Implementing KNN with Cosine Similarity

We convert our table to a 2D matrix, and fill the missing values with zeros (since we will calculate distances between rating vectors). We then transform the values(ratings) of the matrix data frame into a scipy sparse matrix for more efficient calculations.

Finding the Nearest Neighbors, We use unsupervised algorithms with sklearn.neighbors. The algorithm we use to compute the nearest neighbors is "brute", and we specify "metric=cosine" so that the algorithm will calculate the cosine similarity between rating vectors. Finally, we fit the model.

-->Thirdly, we will be implementing an associate rule learning based recommendation system which will use Apriori as the algorithm. So Apriori is about people who bought something, also bought something else.

The apriori algorithm, you can really tell that there are some potential rules that can come out of this, that, for instance, everybody who watches movie one, not everybody, but it is likely that people who watch Movie one will or who like movie one will also like movie number two. And people who like movie number two are quite likely to also like movie number four.

There we can come up with lots of different potential rules, but some are going to be stronger,some are going to be weaker. And we want to find the very strong ones in order to build our business decisions. We can see these things from the data and we want to extract this information.

Association Rule Learning is all about analyzing some things in pairs or in combination together for some reasons/rules.It predicts what customers will buy based on what they bought before. And that's why you see all these suggestions of new products when you buy a certain product. It will be implemented in steps:-

1. Set a minimum support and confidence.
2. Take all the subsets in transactions having higher support than minimum support.

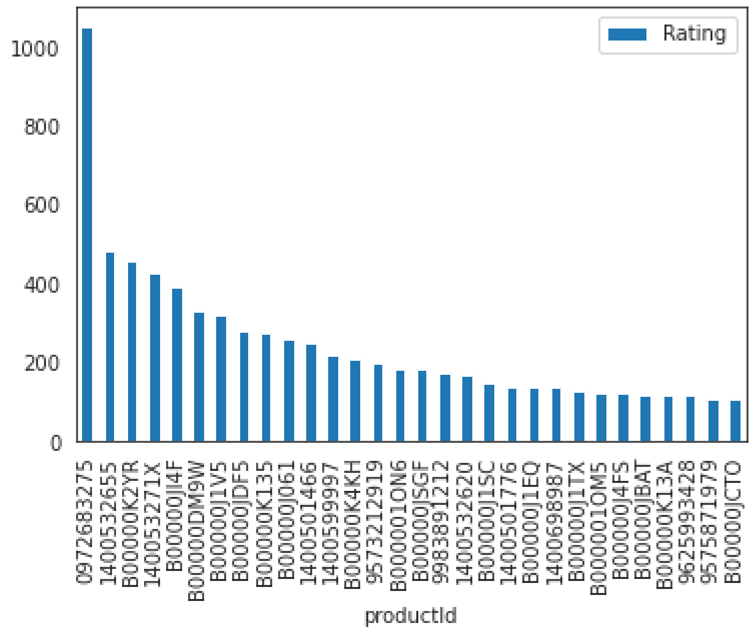
3. Take all the rules of these subsets having higher confidence than minimum confidence.

4. Sort the rules by decreasing lift i.e. the rule of the highest lift given the criteria.

Dataset will contain details regarding 7328 users who purchased some products for themselves either from our website or from our store/warehouse while taking the session about any of the gadgets.

Milestones

1. Popularity based Recommendation System:



2. Collaborative filtering (Item-Item recommendation):

   Recommendations for B00000JYWQ:

   1: B00000JBRP, with distance of 0.9344641908879138:

   2: B00000JBRT, with distance of 0.9489007859243962:

   3: B00000J4A5, with distance of 0.9579073277805836:

   4: B00000JI38, with distance of 0.9623057272473416:
   5: B00000JDKU, with distance of 0.9726780035503074:

3. Association Rule Mining:

Results shows top 7 pairs of products that are mostly associated/correlated with each other are:- (if a person buys product on the left, he will mostly be interested in buying the right one product)

Left Hand Side	Right Hand Side	Support	Confidence	Lift
Smartphones	Wireless Headphones	0.005988	0.400000	5.120557
Laptops	Hard Drives	0.003674	0.259615	4.641494
Full HD Smart LED TV	Pen-Drives	0.007485	0.269608	4.412201
Smartphones	Wireless Bluetooth Earbuds	0.004219	0.281818	4.208943
Split Air-Conditioners	Air Purifiers	0.004627	0.369565	3.993478
Tablets	Wireless Headphones	0.005580	0.303704	3.887831
Air Coolers	Air Purifiers	0.013473	0.295522	3.193380

Feature Selection for Predicting Price Category of Smartphone:

Specifications

So, coming to our in-store sessions that have been taken by techies of Tech_novan, it seems our techies are also data scientists and they will also use some feature-selection techniques to recommend you products based upon some important features which have more relation with the dependent variable which will help to build a good model. Adding unnecessary features while training the model leads us to reduce the overall accuracy of the model, increase the complexity and decrease the generalization capacity of the model.

For demonstration purpose, we are taking an example of a smartphone dataset which will have like 20 features like RAM, the number of cores, 4G supportable, battery power, clock speed, weight, and a lot more to predict the price category which then will be matched by our customer budget for the recommendation of the best product. But in doing so, we are interested in taking the best features out of these 20 features which will help to simplify our dataset and reduce training time for a prediction. The dataset contains around 2000 entries of smartphones with the same or different features with the respective price category as the dependent variable.

We will do data pre-processing that includes data cleaning, splitting of data, feature scaling, and encoding the categorical data. After that, we will start with the building of machine learning algorithms. For this part, since our data has twenty features for getting the dependent variable, feature selection is necessary for this task.

We are going to use all three types of feature selection methods i.e filter methods, wrapper methods, and Embedded Methods. We will kind of take a superset of important features from all three of them. From Filter methods of feature selection, we will implement Information Gain, Correlation Coefficient, and Chi-Square Test. From wrapper methods of feature selection, we will not use any technique since scikit learn will take care of it automatically. From embedded methods of feature selection, we will use ExtraTreeClassifier.

We will use twelve types of classification techniques which include Logistic Regression, KNearestNeighbour Classification, Naive Bayes Classification, Decision Tree Classification, Random Forest Classification, Support Vector Machine Classification, XGBoost Classification, Gradient Boosting Classification, AdaBoost Classification, LightGBM Classification, Catboost Classification, and a Deep Learning Neural Network with two hidden layers. To improve the model performance, we will also apply model selection techniques like K-Fold Cross-Validation and Hyperparameter Tuning using Randomized Search and Grid Search. In the end, we will also compare these models by comparing their accuracies to get the best model. This dataset contains details of 2000 smartphones with 20 features:-

Independent Variable:battery_power,blue,clock_speed,dual_sim,fc,four_g,int_memory,m_dep,mobile_wt, n_cores,pc,px_height,px_width,ram,sc_h,sc_w,

talk_time,three_g,touch_screen,wifi

Dependent Variable:price_range

Description of variables in the above file

battery_power: Total energy a battery can store in one time measured in mAh

blue: Has Bluetooth or not

clock_speed: the speed at which microprocessor executes instructions dual_sim: Has dual sim support or not

fc: Front Camera megapixels

four_g: Has 4G or not

int_memory: Internal Memory in Gigabytes

m_dep: Mobile Depth in cm

mobile_wt: Weight of the mobile phone

n_cores: Number of cores of the processor

pc: Primary Camera megapixels

px_height

Pixel Resolution Height

px_width: Pixel Resolution Width

ram: Random Access Memory in MegaBytes

sc_h: Screen Height of mobile in cm

sc_w: Screen Width of mobile in cm

talk_time: the longest time that a single battery charge will last when you are

three_g: Has 3G or not

touch_screen: Has touch screen or not

wifi: Has wifi or not

price_range: This is the target variable with a value of 0(low cost), 1(medium cost), 2(high cost), and 3(very high cost).

If we take superset of these all these feature-selection techniques, we can take these 6 features for classification task:- 1.ram 2.battery_power 3.px_height 4.px_width 5.mobile_wt 6.int_memory

Milestones

Results Obtained:-

1. Logistic Regression:-

Accuracy: 97.80 %

Standard Deviation: 0.60 %

```
  precision    recall  f1-score   support


0      0.98      0.99      0.99      124
1      0.96      0.98      0.97      110
2      0.97      0.95      0.96      125
3      0.99      0.97      0.98      141
```

2. KNearestNeighbour Classification:-

Accuracy: 87.00 %

Standard Deviation: 2.48 %

```
  precision    recall  f1-score   support


0      0.93      0.98      0.95      124
1      0.82      0.85      0.84      110
2      0.78      0.86      0.82      125
3      0.99      0.84      0.91      141
```

3. SVM Classification:-

Accuracy: 97.07 %

Standard Deviation: 1.08 %

```
  precision    recall  f1-score   support
```

```
0      0.98     0.99     0.98      124
1      0.94     0.97     0.96      110
2      0.97     0.94     0.95      125
3      0.99     0.97     0.98      141
```

4. Naive Bayes Classification:-

```
Accuracy: 79.27 %
Standard Deviation: 2.61 %
  precision    recall  f1-score   support


0      0.93     0.93     0.93      124
1      0.74     0.74     0.74      110
2      0.73     0.72     0.73      125
3      0.90     0.91     0.91      141
```

5. Decision Trees Classification:-

```
Accuracy: 87.00 %
Standard Deviation: 3.43 %
  precision    recall  f1-score   support


0      0.97     0.93     0.95      124
1      0.80     0.88     0.84      110
2      0.78     0.79     0.79      125
3      0.92     0.87     0.89      141
```

6. Random Forest Classification:-

```
Accuracy: 90.47 %
```

Standard Deviation: 1.55 %

```
  precision   recall  f1-score   support


0     0.96     0.98     0.97      124
1     0.91     0.87     0.89      110
2     0.85     0.89     0.87      125
3     0.95     0.92     0.94      141
```

7. XGBoost Classification:-

Accuracy: 90.53 %

Standard Deviation: 2.34 %

```
  precision   recall  f1-score   support


0     0.96     0.98     0.97      124
1     0.93     0.91     0.92      110
2     0.90     0.86     0.88      125
3     0.92     0.95     0.94      141
```

8.AdaBoost Classifier

Accuracy: 73.93 %

Standard Deviation: 3.18 %

```
  precision   recall  f1-score   support


0     0.88     0.84     0.86      124
1     0.61     0.72     0.66      110
2     0.68     0.45     0.54      125
3     0.77     0.94     0.85      141
```

9.Gradient Boosting Classifier

Accuracy: 89.67 %

Standard Deviation: 2.43 %

```
  precision    recall  f1-score   support

0      0.97      0.98      0.98       124
1      0.88      0.89      0.89       110
2      0.86      0.86      0.86       125
3      0.95      0.94      0.94       141
```

10.LightGBM Classifier

Accuracy: 90.53 %

Standard Deviation: 2.21 %

```
  precision    recall  f1-score   support

0      0.95      0.99      0.97       124
1      0.92      0.87      0.90       110
2      0.87      0.87      0.87       125
3      0.94      0.94      0.94       141
```

11. CatBoost Classification:-

Accuracy: 93.27 %

Standard Deviation: 2.16 %

```
  precision    recall  f1-score   support

0      0.98      0.99      0.98       124
1      0.95      0.95      0.95       110
2      0.95      0.92      0.93       125
3      0.97      0.98      0.97       141
```

12. Deep Learning Neural Network:-

Accuracy: 97.2%

	precision	recall	f1-score	support
0	0.98	0.99	0.98	124
1	0.95	0.96	0.96	110
2	0.97	0.95	0.96	125
3	0.99	0.98	0.98	141

So, as we can see, the Logistic Regression and Deep-Learning model have given the best accuracies which is 97.8 and 97.2 along with good precision, recall values. It is clearly visible our ML model can correctly predict with 97% accuracy for the new set of customers which means 97 out of 100 times our techies can suggest a correct set of features for the particular price category.

Conclusion

This is one example of how Data science and Machine learning could be implemented in an e-commerce company which is an online hub for electronic gadgets and computer products. Through smart bots, customer segmentation, recommender system, ad optimization, the company can achieve increased revenue, improve customer retention and provide a seamless end to end experience for the customers. Similar applications can be extended to varied fields and industries to create custom models for specific use cases fulfilling clients' vision.