The Automated Travel Agent: Hotel Recommendations Using ML

...

Mentor Name- Tarun Gupta
Project Guide- Dr. Avinash Sharma

Kushagra Chandak
Diplav Srivastava
Harshil Jain

Dataset Information

Data Source - Kaggle

Number of Instances - 4 million

Number of features -22

Problem Statement

Recommending best hotel to customer on the basis of some parameters and hence giving a personalized hotel recommendation to all users.

Expedia is interested in predicting which hotel group a user is going to book.

In-house algorithms to form hotel clusters

These hotel clusters serve as good identifiers to which types of hotels people are going to book, while avoiding outliers such as new hotels that don't have historical data

Feature Explanation

There are many features given like user location,hotel location,hotel rating,distance between origin and destination etc.

All these features play a significant role while booking a hotel like if the hotel lies in different continent with other features similar has less probability compared to other hotel within the same continent or state.

Similarly hotel rating and family specifications matter for making a recommendation better.

Features

Feature name	Description	Data type
site_name	ID of the Expedia point of sale (i.e. Expedia.com, Expedia.co.uk, Expedia.co.jp, ...)	int
posa_continent	ID of continent associated with site_name	int
user_location_country	The ID of the country the customer is located	int
user_location_region	The ID of the region the customer is located	int
user_location_city	The ID of the city the customer is located	int
orig_destination_distance	Physical distance between a hotel and a customer at the time of search. A null means the distance could not be calculated int	double
is_mobile	1 when a user connected from a mobile device, 0 otherwise	int
is_package	1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise	int
channel	ID of a marketing channel	int
srch_adults_cnt	The number of adults specified in the hotel room	int
srch_children_cnt	The number of (extra occupancy) children specified in the hotel room	int
srch_rm_cnt	The number of hotel rooms specified in the search	int
srch_destination_id	ID of the destination where the hotel search was performed	int
srch_destination_type_id	Type of destination	int
hotel_continent	Hotel continent	int
hotel_country	Hotel country	int
hotel_market	Hotel market	int

Feature Extraction and Selection

Most of the features are specified correctly but some new features are introduced from the current features like in_time and out_time from a hotel is given.We have converted into length_of__stay which is a more significant or a useful feature for our problem.

Algorithms Implemented

- Naive Bayes
- User Similarity Kernel
- SVM(RBF Kernel)

Naive Bayes

...

Pre Processing Done

Replaced Missing attribute *with their Mean*

We filter out all training examples where the user did not actually book the hotel cluster that they clicked on and replaced all the example that contained missing information with their mean.

Downsampling

Since the training dataset was so huge(30 million data point) and required a lot computational time , we downsampled the training dataset such that prior probability of all the hotel cluster remained same

More useful feature obtained

Some of the feature like date in time and out time was converted into length_of__stay which is a more significant or a useful feature for our problem

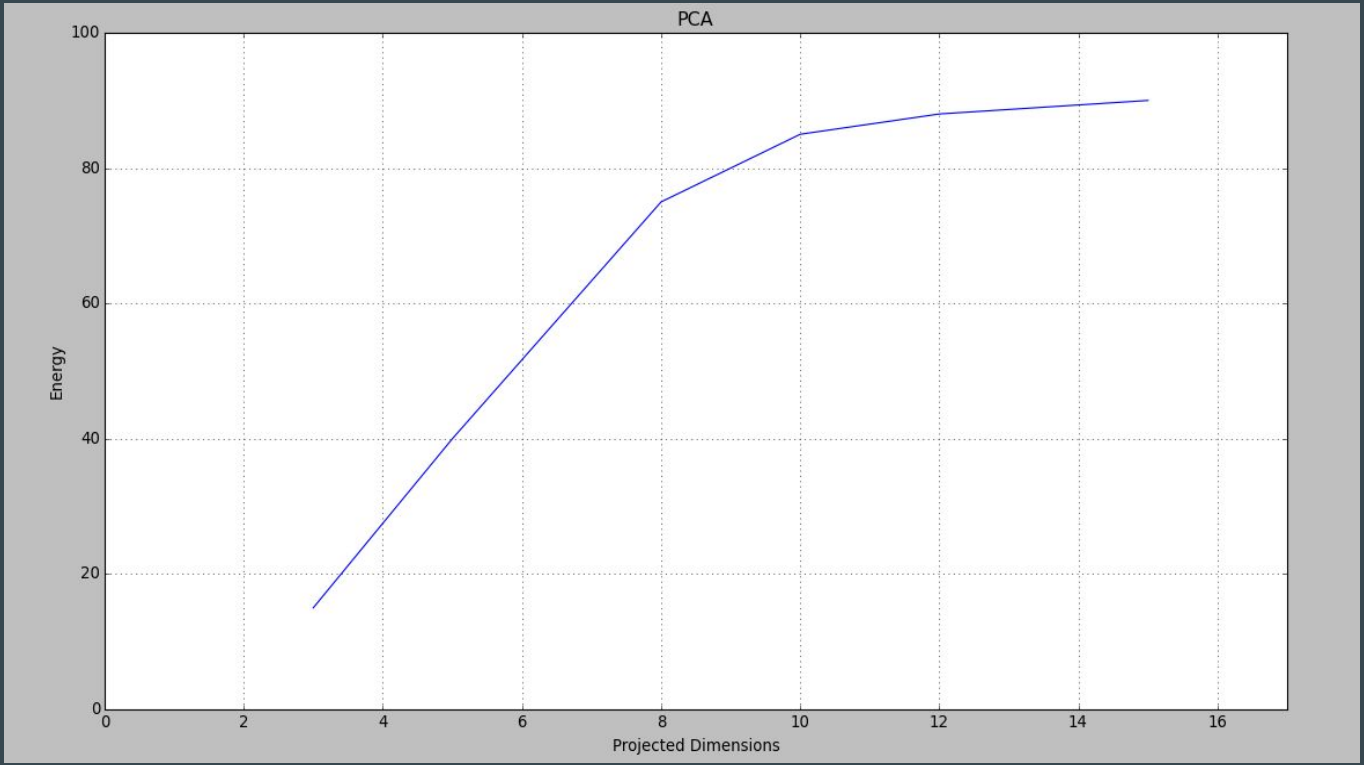
Performed PCA

Performed PCA on the resultant training dataset to reduce the number of attribute from 24 to 12

Analysis of Naive Bayes

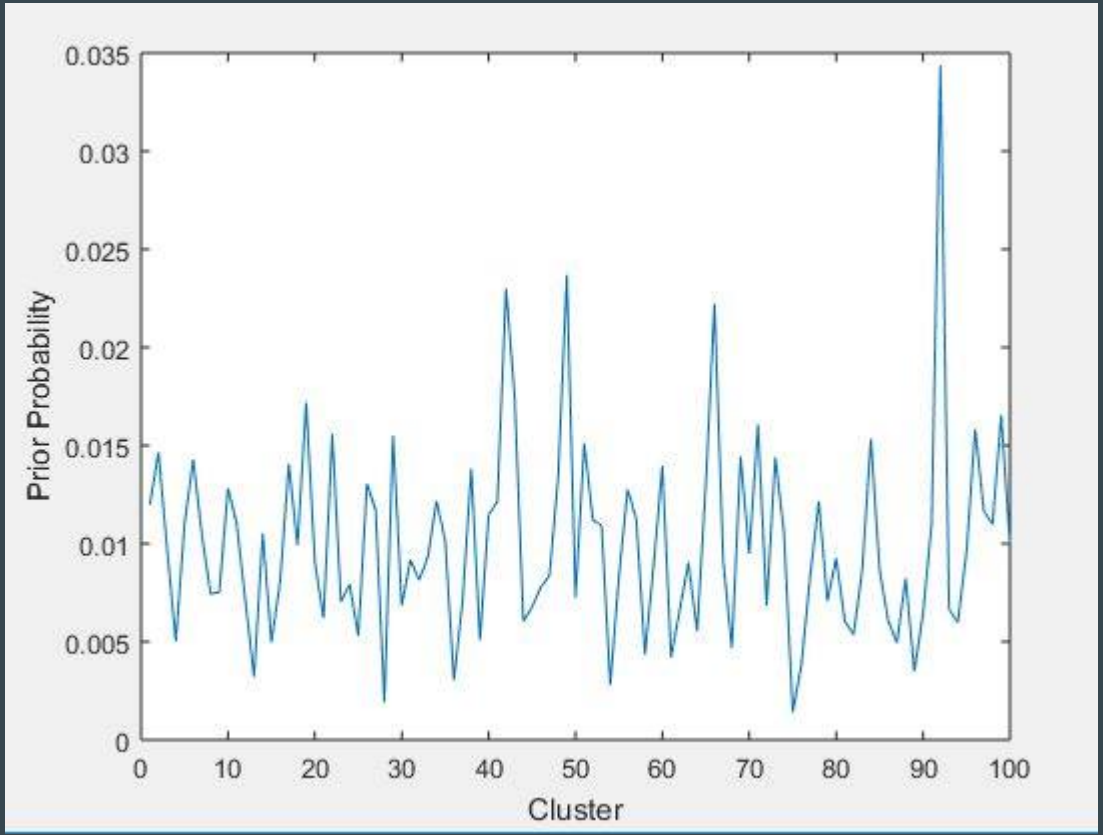
For Naive Bayes, we hand-selected a few features, to get a basic machine learning algorithm running.We calculated target class probabilities for the categorical features, and we normalized continuous variables based on target label values. For each attribute in the test set, we converted its value to booking probability based on the the observed probabilities in the training set.

Choosing the value of K(Projected Dimension)



$$\text{Energy} = \frac{\left(\sum_{i=1}^{K} \lambda_i\right)}{\left(\sum_{j=1}^{N} \lambda_j\right)}$$

Prior Probability of Cluster



Assumptions

Attribute Linearly Independent

The main assumption made while applying Naive Bayes classifier was that attribute of training dataset obtained after PCA are linearly independent so that we can directly treat each attribute as independent random variable and multiply each of them to get probability of each cluster

Results

Algorithm	Precision	Recall	F1
Naive Bayes	0.0596321	0.0567834	0.057489

Validation Technique Used

We used holdout technique for validation.

The features used to train the model were generated using only hotel cluster and data prior to '2014-07-01'.

A portion of the training data from '2014-07-01' onward was set aside for validation.

Support Vector Machines(SVM)

...

Support vector machines (**SVMs**]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Support Vector Machine (SVM)

Applied PCA

We have first applied PCA to the data to reduce the size of features which in place reduces the computation a lot . On Applying PCA we reduced the number of features from 22 to 10.

Downsampling

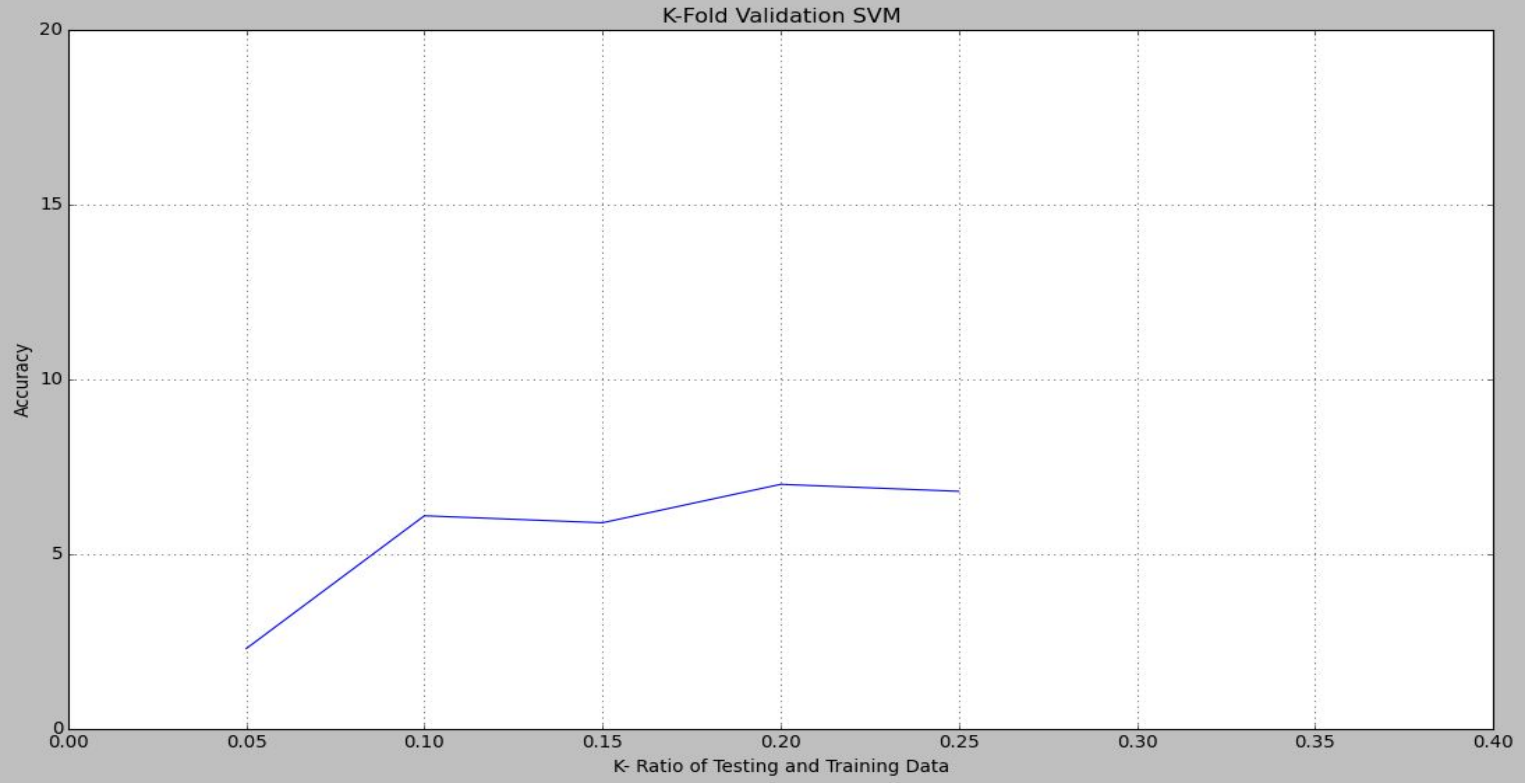
Since the training dataset was so huge(30 million data point) and required a lot computational time , we downsampled the training dataset such that prior probability of all the hotel cluster remained same

Support Vector Machine (SVM)

Changed the value of Gamma and test and train data

Then applied SVM RBF Kernel over the downsampled data and observed the results by varying different values of gamma and ratio of training and testing data(K).

K-Fold Validation(SVM)



Assumptions

After applying PCA ,then Kernel function the training data has become linear upto certain extent.

Results

SVM RBF

Algorithm	Precision	Recall	F1
SVM RBF	0.073425	0.069846	0.071976

SVM Polynomial

Results

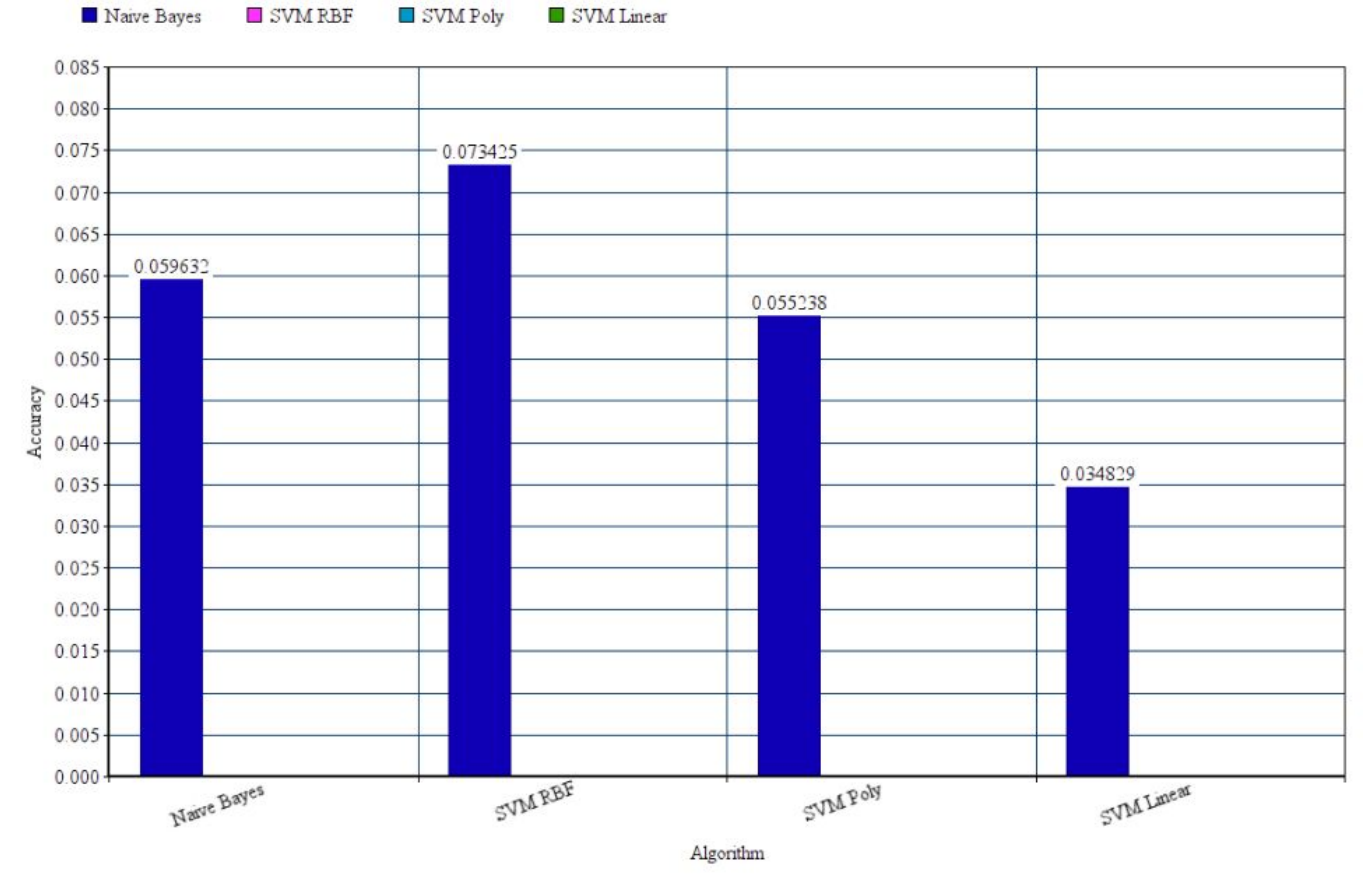
Algorithm	Precision	Recall	F1
SVM Polynomial	0.073425	0.069846	0.071976

Results

SVM Linear

Algorithm	Precision	Recall	F1
SVM linear	0.034829	0.032112	0.034091

Comparison of Algorithm



Problem Challenges

Huge Dataset

Since the training dataset was so huge(30 million data point) and required a lot computational time one of the important task was to downsample data such that most of the useful information is retained

Few attribute were non vectorial

Few feature were of string form but had significant importance , so converting that attribute into integer form.