Mexican tourism recommendation system using Machine Learning models

Juan Pablo Enríquez Pedroza
Facultad de Ingeniería
Universidad Panamericana
Aguascalientes, México
0228903@up.edu.mx

Ulises Gallardo Rodríguez
Facultad de Ingeniería
Universidad Panamericana
Aguascalientes, México
0229261@up.edu.mx

***Abstract*—In this paper, a tourism recommendation system based on several Machine Learning models (Decision Tree, SVC, Logistic Regression, Random Forest) and a technique for processing the data (One Hot Encoder) is presented. The system was focused on a specific problem given in a contest, that is defined as: "Given a TripAdvisor tourist and a Mexican tourist place, the goal is to automatically obtain the degree of satisfaction (between 1 and 5) that the tourist will have when visiting that place".**

Keywords—classification, machine learning, metrics. recommendation.

I. Introduction

Tourism is a social, cultural, and economic phenomenon related to people's movement to places outside their usual residence for personal or business/professional reasons. This activity is vital in various countries, including Mexico, representing 8.7% of the national GDP, generating around 4.5 million direct jobs.

With the pandemic generated by the SARS-COV-2 virus, which began in Mexico in mid-March 2020, tourism was one of the most affected sectors. Tourism is trying to re-establish itself through improvements in the quality and safety of touristic products and services. For this, systems can be developed that consider the user and destination information to recommend the places where the user will have better tourist experiences [7].

To accomplish this, we can rely on Machine Learning techniques. Machine Learning is the “field of study that gives computers the ability to learn without being explicitly programmed”. Arthur Samuel (1959) [6].

II. Methodology

A. Recommendation System corpus

The dataset is divided into three files: the recommendation data test, the Mexican tourist places and a set of history’s opinions in TripAdvisor from each user.

Each instance of the recommendation data set consists of six columns:

1. *Index*: It is the index of each recommendation.
2. *Gender*: The tourist's gender.
3. Place: The tourist place that the tourist is recommended to visit.
4. *Location*: The place of origin of the tourist (regions of Mexico).
5. *Date*: Date the recommendation was issued.
6. *Type*: Type of trip that the tourist would do.

The label that must be predicted ranges from 1, which means the highest degree of dissatisfaction, to 5, which is the highest degree of satisfaction.

Mexican tourist places’s file contains the 18 places that can be recommended. These places are in the state of Nayarit, in Mexico. Each line contains the name of the place and its description with the most important information about it.

Finally, the tourist's history file consists of a set of opinions of various places from anywhere in the world together with their corresponding satisfaction degrees. These opinions in some way describe the tourist's appreciation of each place visited in the past, correlating with the rating value (tourist's satisfaction degree) given to that place [1].

B. Related work

We selected two related papers that try to solve this problem [8]:

- *An Embeddings Based Recommendation System for Mexican Tourism. Submission to the REST-MEX Shared Task at IberLEF 2021 (Arreola et al., 2021)*
  – Team: Alumni-MCE 2GEN
  – Summary: The team proposes two methods, the first one is based on Doc2vec. The Doc2Vec model was applied to the user and place information of the dataset. The obtained embeddings were matched with the reviews' centroid embeddings through similarity metrics, and these embeddings were assigned to the design matrix. Finally, for the other user variables, a hot encoding was applied to be incorporated in the design matrix to be modeled through a Neural Network with one hidden layer and ordinal encoding to deal with the unbalanced problem of the data. They proposed a system based on distributed representations of texts for the second method, using the BERT approach.
- *A Recommendation System for Tourism Based on Semantic Representations and Statistical Relational Learning (Morales-González et al., 2021)*
  – Team: Labsemco-UAEM
  – Summary: The team presented a method of text representation different from the methods of lexical co-occurrence in text. This method extracts the linguistic features in the text, specifically the lexical and semantic signals of synonymy-antonymy. They proposed to use the ComplEx model for the recommendation task. The model was modified to predict the target label, considering it as a relationship between a User and a Place.

C. Data preprocessing

During the first stages of the analysis process, we found several problems when processing the data:

- There is no clear match between the review a user made for the Mexican place he visited and the history of reviews of the previous worldwide places he has visited before.
- Each user reviewed just one of the main 18 places of the dataset.
- The reviews history of the users have not a unified language, some are in spanish and other in english, and we couldn't find a free tool to translate them.
- Most of the data is biased.
- In order to take advantage of the reviews history, we should use Natural Language Processing techniques, and we do not have the required knowledge for applying them.

All these problems led us to our final proposal solution, using only Machine Learning techniques. Our proposal is described below.

D. Our Proposal

Our final approach, after several failed attempts, is trying to use the main characteristics from the recommendation data set, transforming nominal categorical features into numerical features with One Hot Encoder when processing the data, in order to train some classification models to predict the satisfaction degree.

E. Techniques

One Hot Encoder

One-hot encoding, otherwise known as dummy variables, is a method of converting categorical variables into several binary columns, where a 1 indicates the presence of that row belonging to that category.

With this methodology we have a total of 75 features in our final dataset that not only does it add a massive number of dimensions to it, there really isn't much information, ones occasionally dotting a sea of zeroes.

We selected and trained several models to find the one that could make the best predictions based on the data we have. The models used from sklearn implementations are:

Decision Tree Classifier

It's a tree structure that defines 'questions' applying rules to make decisions, in order to reduce the impurity of the leaf node on each node. It continually splits the dataset until it isolates all data points belonging to each class.

Support Vector Classifier (SVC)

The *SVC*, described in [9], is a supervised learning model for, mainly, classification problems. But sometimes a linear model is not enough. To improve this, it can take advantage of kernels, which transform the data from the original vectorial space to another vectorial space to easily separate the classes. For our purpose, we use the Radial Basis Function (RBF) Kernel [3], which is the most generalized form of kernelization and is one of the most widely used kernels due to its similarity to the Gaussian distribution. The RBF kernel function for two points X_1 and X_2 computes the similarity or how close they are to each other.

LogisticRegression

Is a supervised learning model for classification problems. It consists of fitting a point (1D), a line (2D), a plane (3D) or a hyperplane (+3D) to separate the data of different classes. [6] There are three types of Logistic regression[5]:

1.- Binary Logistic Regression where the categorical response has only two 2 possible outcomes.

2.- Multinomial Logistic Regression for three or more categories without ordering.

3.- Ordinal Logistic Regression for three or more categories with ordering. such as movie ratings from 1 to 5.

RandomForest.

Is a supervised learning model formed by several decision trees. All the trees are different among them because they are trained with a random subset of samples and features.[6] The prediction of a new observation is obtained by adding the predictions of all the individual trees that make up the model. en scikit-learn es necesario hacer one-hot-encoding para convertir las variables categóricas en variables dummy.[10]

III. Results

After the data preprocessing, the feature selection and the models training, we got the following results:

Table 1: Performance for each model

	Accuracy	F1 score	MAE
Decision Tree	0.484725	0.275586	0.830957
SVC	0.549898	0.260177	0.639511
Logistic Regression	0.533605	0.198428	0.655804
Random Forest	0.558045	0.260334	0.643585

The selected metric for measuring the performance, as shown in the Table 1, are:

Accuracy.

It calculates the percentage of correct predictions. Its values range between 0 and 1, where 1 represents that all the test samples were predicted correctly.

F1.

The F1 score can be interpreted as a weighted average of the precision and recall values, where an F1 score reaches its best value at 1 and worst value at 0.

Mean Absolute Error.

MAE is the average of all absolute errors. This metric is oriented to regression models and it's added because it was the metric selected to evaluate the contest.

The reason for the selection of the previous metric was because, first, they tell us a lot about the real performance of the selected models and, second, those metrics were the ones being evaluated by the judges of the contest we originally were focusing on as our actual solution.

Analyzing our results (Table 1) we can observe:

- SVC is the best model on average.
- The second best model on average is the Random forest.
- Decision tree is the worst model in both accuracy and MAE, but is the best in F1 score. If we discard this last metric, the Decision tree is the model that gave us the best predictions.

- Logistic regression is the worst model based on the F1 score; it is almost 7 points under the second worst score in this metric: SVC.

Compared with the metrics that the other contestants had in their models (Table 2), we could be over the Labsemco-UAEM Run 1 model in the MAE metric with our SVC results. Also, our best F1 result, obtained by the Decision Tree, would be the 7th best result in the contest, as well as with the MAE metric.

Table 2: Performance for the Recommendation System task competition.

Team	MAE	F-measure	Accuracy
Minería UNAM$_{Run1}$	**0.47**	0.42	**56.72**
UCT-UA$_{Run2}$	0.54	**0.45**	53.24
UCT-UA$_{Run1}$	0.56	0.40	53.83
DCI-UG$_{Run1}$	0.56	0.28	53.33
Minería UNAM$_{Run2}$	0.58	0.24	54.78
DCI-UG$_{Run1}$	0.60	0.25	53.70
Labsemco-UAEM$_{Run1}$	0.64	0.30	49.05
Techkatl$_{Run1}$	0.66	0.27	50.18
Baseline	0.72	0.13	51.35
Arandanito Team	0.76	0.16	45.71
TextMin-UCLV*$_{Run1}$	0.78	0.17	36.23
Techkatl$_{Run2}$	0.81	0.21	44.76
Labsemco-UAEM$_{Run2}$	0.91	0.24	36.50
TextMin-UCLV*$_{Run2}$	1.00	0.18	38.31
The last	1.26	0.21	36.95

IV. Conclusion

This paper presented our proposal of a tourism recommendation system using a set of four classification models trained with our processed data and measured with some metrics that allowed us to understand the performance of each model, in order to know which was the best for this problem.

After all this analysis of the dataset and our results, we found that our models do not give good results, neither do the models made by the other contestants. This is due to the biased data given in the dataset and, in our case, the lack of knowledge to apply more advanced techniques like Natural Language Processing to take advantage of other of the given datasets to try to improve a little the performance of our models.

V. References

[1]Morales-González, E., D. Torres-Moreno, A. Ehrlich-Lopez, M. Toledo-Acosta, B. Martnez-Zaldivar, and J. Hermosillo-Valadez. 2021. A recommendation system for tourism based on semantic representations and statistical relational learning. In Proceedings of the Third Workshop for Iberian Languages Evaluation Forum (IberLEF 2021), CEUR WS Proceedings.

[2] Arreola, J., L. Garcia, J. Ramos-Zavaleta, and A. Rodríguez. 2021. An embeddings based recommendation system for mexican tourism. submission to the rest-mex shared task at iberlef 2021. In Proceedings of the Third Workshop for Iberian Languages Evaluation Forum (IberLEF 2021), CEUR WS Proceedings.

[3]Sreenivasa, S. (2020, October 12). *Radial basis function (RBF) kernel: The go-to kernel*. Medium. Retrieved June 5, 2022, from https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a

[4]Bento, C. (2021, July 18). *Decision tree classifier explained in real-life: Picking a vacation destination*. Medium. Retrieved June 7, 2022, from https://towardsdatascience.com/decision-tree-classifier-explained-in-real-life-picking-a-vacation-destination-6226b2b60575

[5]Swaminathan, S. (2019, January 18). *Logistic regression - detailed overview*. Medium. Retrieved June 7, 2022, from https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

[6]"Introduction to Machine Learning" (Claudia Nalley Sánchez Gómez, personal communication, June 06, 2022).

[7] IberLEF. (2022, January 15). REST-MEX: Recommendation System for Text Mexican Tourism. REST-MEX. Retrieved June 7, 2022, from https://sites.google.com/cicese.edu.mx/rest-mex-2021

[8] Alvarez Carmona, M. Á., Aranda, R., Arce Cardenas, S., Fajardo Delgado, D., Guerrero Rodríguez, R., López Monroy, A. P., Martínez Miranda, J., Rodríguez González, A. Y., & Pérez Espinosa, H. (2021). Overview of Rest-Mex at IberLEF 2021: Recommendation System for Text

Mexican Tourism. Sociedad Española Para El Procesamiento Del Lenguaje Natural, 67, 163–172. http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6386

[9] Heras, J. M., & Portella, J. G. V. (2019, May 28). *Máquinas de vectores de Soporte (SVM)*. IArtificial.net. Retrieved June 7, 2022, from https://www.iartificial.net/maquinas-de-vectores-de-soporte-svm/

[10]Amat Rodrigo, J. (2020, October). *Random Forest con python*. Random Forest python. Retrieved June 7, 2022, from https://www.cienciadedatos.net/documentos/py08_random_forest_python.html

Meter algo de PLN

Hacer más formal el documento