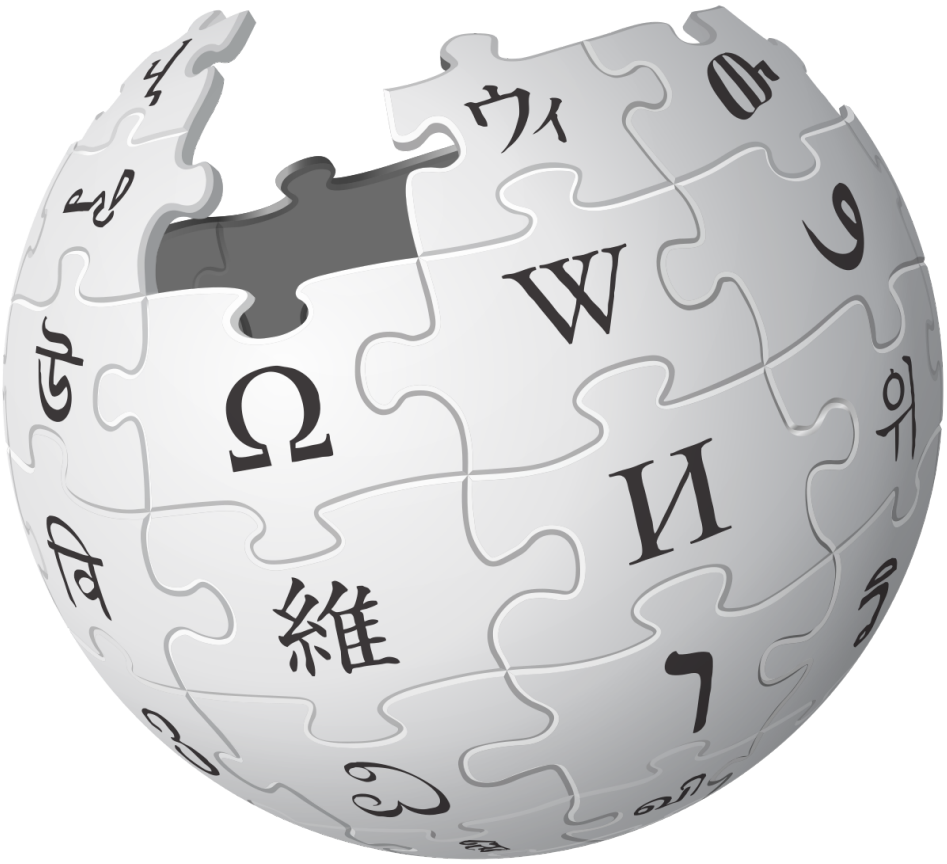
ITCS 3190 Final Project Report:

Wikipedia Learning Approach



Abel Villanueva

May 3, 2022

Table of Contents:

Introduction

With the ever-increasing information website, Wikipedia has been greatly increasing over the years but still may be guarded over its' accuracy given that anyone can edit a page. Its use in an academic setting has been up for debate as a trustworthy citation to use in a paper.

My name is Abel Villanueva and I am a student at UNCC, double majoring in Computer Science with a concentration in Data Science and Mathematics with a concentration in Statistics. While I was looking for any dataset that seemed interesting at https://archive.ics.uci.edu/ml/index.php, I came across wiki4HE.csv which seemed more interesting than others that were there and simple to choose the features for processing. There were no major deviations from the project proposal. The major question to be answered is the (perceived) effectiveness Wikipedia has on students' learning and/or teachers' teaching.

Dataset

The data set used is from a questionnaire given to University faculty on the use of Wikipedia as a teaching resource. The survey required the respondents to rate statements regarding the perceived usefulness, quality, etc. of Wikipedia on a Likert Scale (1-5). Based on a Technology Acceptance Model, the following statements were asked in the survey:

PU1: The use of Wikipedia makes it easier for students to develop new skills

PU2: The use of Wikipedia improves students' learning

PU3: Wikipedia is useful for teaching

QU4: In my area of expertise, Wikipedia has a lower quality than other educational resources

VIS1: Wikipedia improves the visibility of students' work

SA3: It is important that students become familiar with online collaborative environments

USE1: I use Wikipedia to develop my teaching materials

USE2: I use Wikipedia as a platform to develop educational activities with students

USE3: I recommend my students to use Wikipedia

USE4: I recommend my colleagues use Wikipedia

JR2: My university considers the use of open collaborative environments on the Internet as a teaching merit

BI1: In the future I will recommend the use of Wikipedia to my colleagues and students

INC4: To design educational activities using Wikipedia, it would be helpful: greater institutional recognition

EXP1: I consult Wikipedia for issues related to my field of expertise

Preprocessing

The dataset contains many categorical data that I had to look through before trying to train and fit a Machine Learning model. I used pandas to fit a data frame of the data and look at what features have lots of missing data and remove them from the dataset. I also remove data containing information regarding the faculty member's professional career since we are not interested in that information.

There are two features of interest in the dataset, PU2, and PU3 that I considered basing the predictor on. I used scikit's SelectKBest and chi2 functions to select the categorical variables that are of use to reduce the curse of dimensionality when processing. From the plot, I saw that PU3 gives better scores overall than PU2 and I selected the top 13 scores (score>50). We could easily increase or lower the range but this seems to be a decent fit for now. Surprisingly the age and the years of experience of the faculty member did not cut while the ones stated in the Dataset section did.

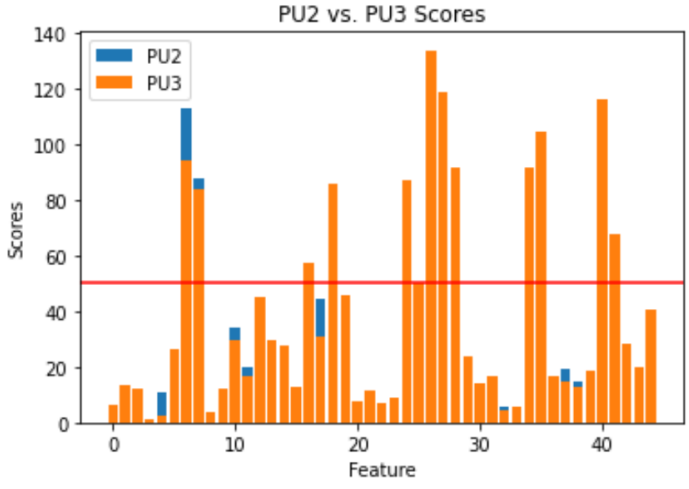


Figure 1: Plot comparing scores of PU2 and PU3 against the rest of the features in the data

Finally, I needed to split our data into our training and testing sets for processing purposes and use scikit's train_test_split for an 80:20 split of our data.

Processing

There is a multitude of classification models on scikit that I could have used to train our model. I decided to use the following and compare their metrics: DecisionTreeClassifier, LogisticRegression, SVC, GaussianNB, MultinomialNB, KNeighborsClassifier, RandomForestClassifier, and GradientBoostingClassifier. We fit the training data onto the model and predict based on the testing data. We calculate accuracy, f1 score, and mean absolute error (MAE) for each model, and a result is shown in Figure 2. MAE is one of the many error metrics for summarizing and assessing the quality of machine learning models. MAE helps in determining the difference between the actual vs. predicted value. In simple terms, it is the average of the residuals in the data.

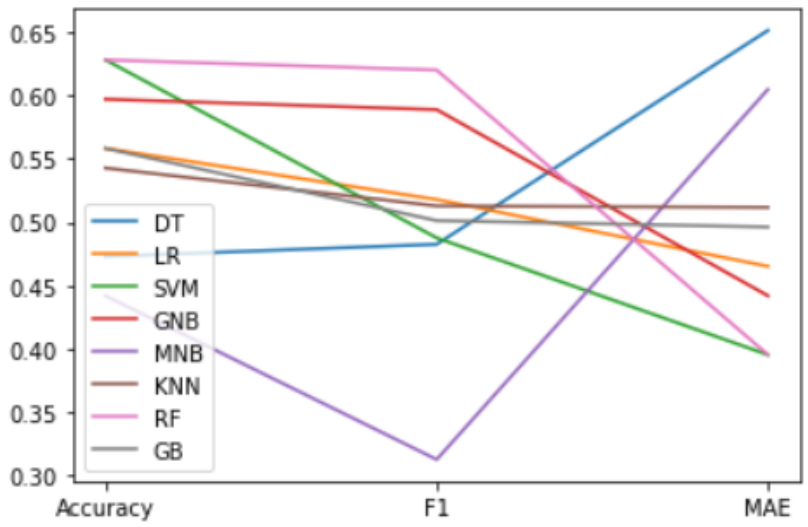


Figure 2: Output Comparing Different Classification Models

Given that the training and testing sets are randomly chosen this would yield different results but for this case, we select the Random Forest Model given that it has high accuracy and f1 scores and a low MAE value compared to the others. For example, Figure 3 on a different test

gives different results and we would most likely choose Gradient Boosting. Figures 2 and 3 illustrate the many models, which all performed uniquely.

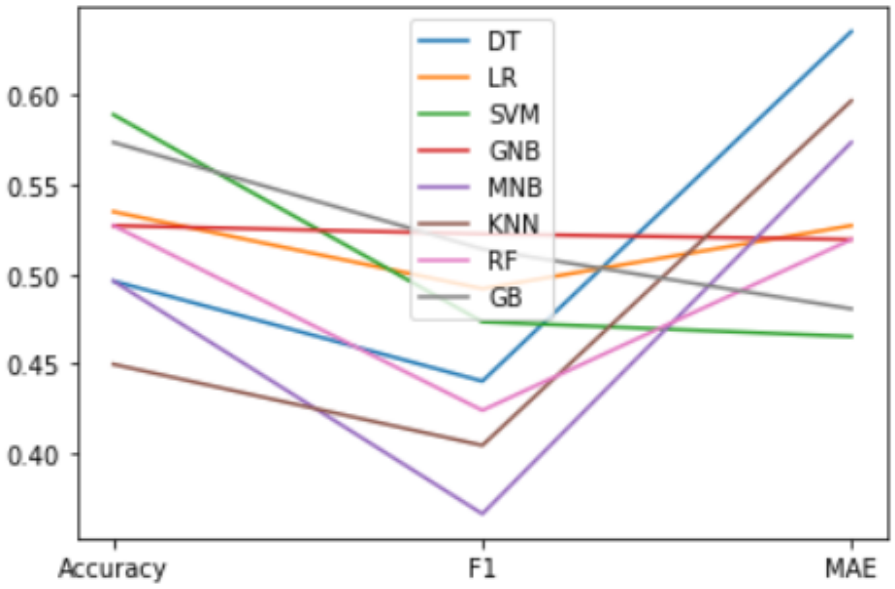


Figure 3: Another Output Comparing Different Classification Models

The results, after running the program multiple times, have shown that Random Forest, Multinomial Naive Bayes, and Gradient Boosting are the prominent models for this dataset. For better performance of these models, we would want the models to have their MAE scores reduced to lower numbers.

AWS Usage

Though I am foreign to working with Amazon Web Services, I was able to quickly get over the learning curve and figured that I needed to utilize a bucket in storing my data. Especially, I was able to successfully store the data in an S3 bucket to make it easier to load the data to train the model. Figure 4 contains the CSV file and checkpoints. I choose this service in particular because I figured this is the most accessible and efficient service for my testing and training data.

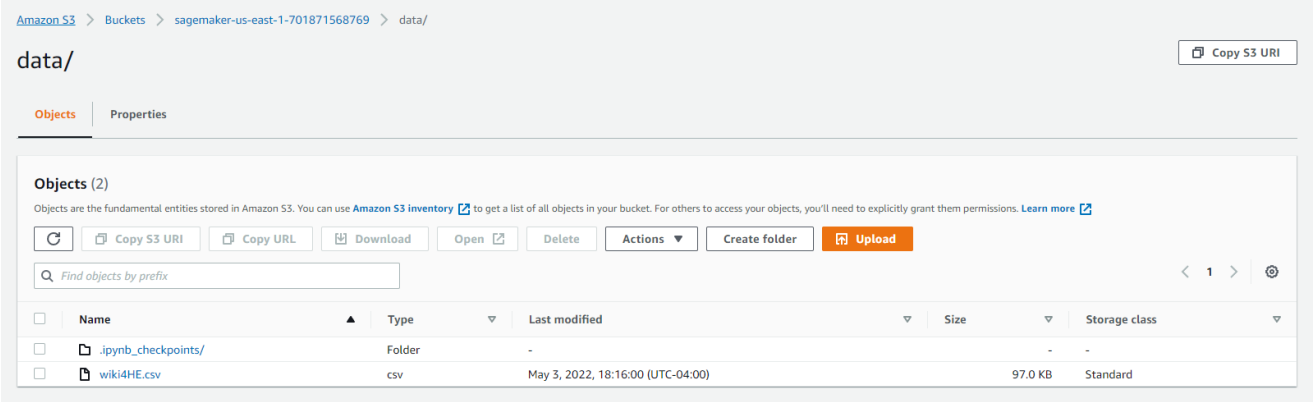


Figure 4: Simple Cloud Storage of a Buckets used

Additionally, I also utilized the AWS SageMaker to make it possible to store a Jupyter Notebook (Figure 5) and run the code necessary to train our model. The process itself was lengthy, but after many hours of debugging, I was able to get it trained and tested (Figure 6). I also used SageMaker to deploy it and make predictions on input data.

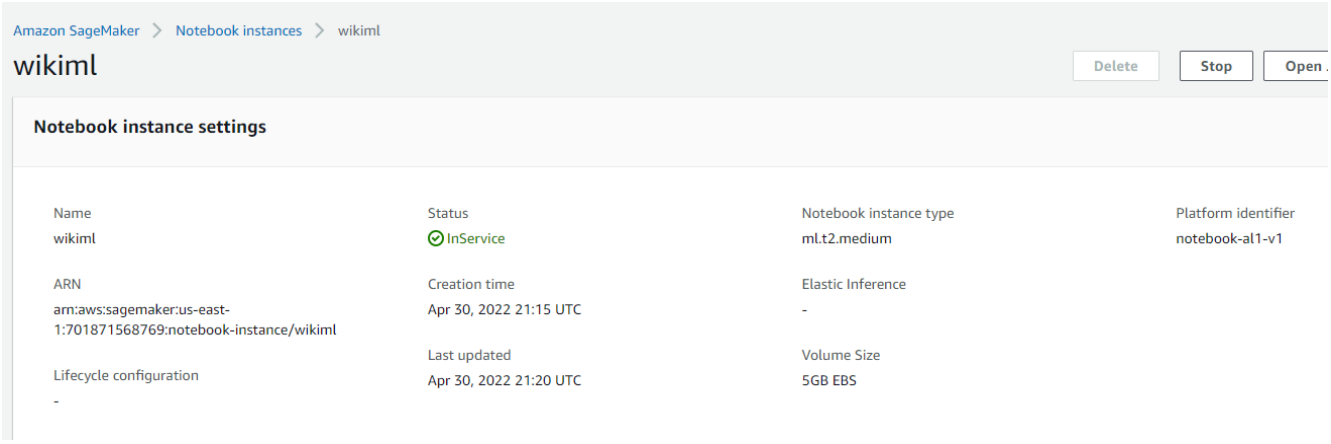


Figure 5: Jupyter Notebook in AWS SageMaker

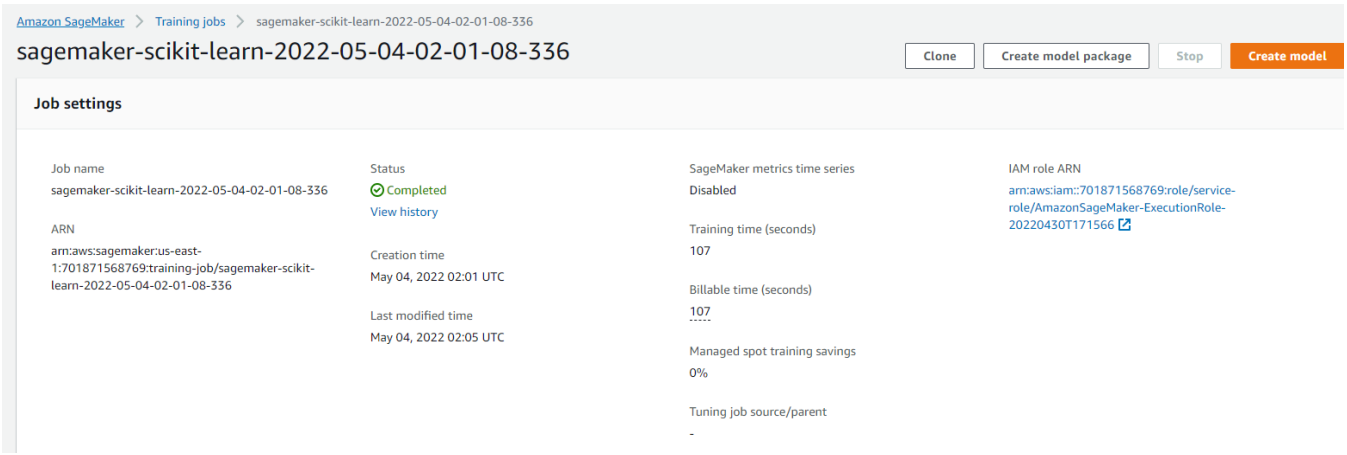


Figure 5: Completed Training Model in AWS SageMaker

Challenges

In the project some difficulties arose, the first being the dataset that I chose. The dataset contained only 913 responses and already had missing values in it. This meant that I may not have had enough data to give a reasonable model for the question at hand. The dataset also only contained faculty members from two universities and both of them are based in Barcelona, Spain. This could mean that we do not have an accurate representation of professors in a larger or different setting. In the future, if I were to continue this research I would ask a different range of universities and get more faculty members. I also did not have as much connection with the topic/dataset as I wished I did which may have affected my investment and overall effort that I contributed to the project.

Another struggle that I had in the project was during processing where I had to heavily reference the scikit manual since I have not used it in a while. I also feel that I had difficulties in choosing the right parameters for each model to get the best output for the model. I hope that in the future I could better understand the differences between the different models so that I could choose one faster.

Throughout the project, my biggest concern regarding it is making use of AWS to the extent that it could. I felt that I did not make full use of all of Amazon Web Services and felt that I could also use other services but did not have the chance to implement them such as an EC2 instance for more computing power or Lambda for our model prediction. I also had difficulties training the model in SageMaker shown by Figure 7's attempts at doing so.

Name	Creation time	Duration	Status
sagemaker-scikit-learn-2022-05-04-02-01-08-336	May 04, 2022 02:01 UTC	4 minutes	Completed
sagemaker-scikit-learn-2022-05-04-01-23-17-430	May 04, 2022 01:23 UTC	4 minutes	Failed
sagemaker-scikit-learn-2022-05-04-01-13-10-456	May 04, 2022 01:13 UTC	4 minutes	Failed
sagemaker-scikit-learn-2022-05-04-00-58-43-732	May 04, 2022 00:58 UTC	4 minutes	Failed
sagemaker-scikit-learn-2022-05-04-00-51-00-973	May 04, 2022 00:51 UTC	3 minutes	Failed
sagemaker-scikit-learn-2022-05-04-00-46-40-004	May 04, 2022 00:46 UTC	4 minutes	Failed
sagemaker-scikit-learn-2022-05-04-00-38-01-278	May 04, 2022 00:38 UTC	5 minutes	Failed
sagemaker-scikit-learn-2022-05-04-00-18-57-278	May 04, 2022 00:18 UTC	3 minutes	Failed
sagemaker-scikit-learn-2022-05-03-23-54-02-247	May 03, 2022 23:54 UTC	4 minutes	Failed
sagemaker-scikit-learn-2022-05-03-23-38-09-565	May 03, 2022 23:38 UTC	4 minutes	Failed

Figure 7: Multiple Attempts at Training the Model