

Team Iowa - Project 2

Project Name

Sentiment Analysis of Steam Game Reviews

Participants

Andre Sharp | 11374558 | AndreSharp@my.unt.edu

Dan Waters | 11457837 | DanWaters@my.unt.edu

Bryan Adams | 10976617 | BryanAdams@my.unt.edu

Haiyi Wang | 11528159 | HaiyiWang@my.unt.edu

Sima Siامي-Namini | 11447869 | SimaSiامي-Namini@my.unt.edu

Workflow

We will use Discord, email, and Zoom meetings to connect with each other. We will use GitHub as the code version control. We can update the code and version on GitHub.

Abstract

Sentiment analysis is one of the major topics in Natural Language Processing (NLP) and Text Mining. The purpose of this project is to use sentiment analysis for game reviews of the Steam platform. The sentiment analysis will be processed from data gathering and data preparation to final classification on a user-generated sentiment dataset. The [Sentiment Analysis for Steam Games](#) dataset from Kaggle will be used. The performance and effectiveness of different classification algorithms such as Auto ML text classification, Vertex AI text classification, Logistic Regression, Random Forest, KNN, neural network(word2vec). This project develops a software for sentiment prediction, and implements text highlighting for explanation of predicted class. The trained classifier can be used to make predictions for unlabeled reviews. The software is used to perform experiments with different parameters,

classification algorithms and visualize how words of a review are influencing the predicted sentiment label.

Index Terms: Sentiment Analysis, Feature Selection, Supervised Machine Learning, Text mining, Natural Language Processing, Classification Model

Background and Motivation

Understanding general sentiment about a topic has long been an important part of product development and marketing cycles. Until recently, however, the ability to collect, process, and analyze sentiment at scale has been rather limited. Organizations and brands have run surveys, conducted brand lift studies, and relied on reviews from professional outlets (such as Consumer Reports) to extract feedback about products and services.

Now, any given individual has a choice of platforms where they can make their thoughts known in a public venue, such as Twitter, Amazon.com, or really, anywhere that users can post. Call center conversations can be converted to text and analyzed by the thousands, as with chatbot interactions.

In this project, we will explore the topic of sentiment analysis, compare and contrast different mechanisms for implementing it, and deliver a demo application that conducts sentiment analysis in real time with user-provided text.

Dataset

For this project, we will be using the [Sentiment Analysis for Steam Games](#) dataset from Kaggle. This data contains over 17,000 rows and the following columns:

			feature	target
review_id	title	year	user_review	user_suggestion

```
In [3]: 1 df_train = pd.read_csv('train.csv')
        2 df_train
```

Out[3]:

	review_id	title	year	user_review	user_suggestion
0	1	Spooky's Jump Scare Mansion	2016.0	I'm scared and hearing creepy voices. So I'll...	1
1	2	Spooky's Jump Scare Mansion	2016.0	Best game, more better than Sam Pepper's YouTu...	1
2	3	Spooky's Jump Scare Mansion	2016.0	A littly iffy on the controls, but once you kn...	1
3	4	Spooky's Jump Scare Mansion	2015.0	Great game, fun and colorful and all that.A si...	1
4	5	Spooky's Jump Scare Mansion	2015.0	Not many games have the cute tag right next to...	1
...
17489	25535	EverQuest II	2012.0	Arguably the single greatest mmorp that exists...	1
17490	25536	EverQuest II	2017.0	An older game, to be sure, but has its own cha...	1
17491	25537	EverQuest II	2011.0	When I frist started playing Everquest 2 it wa...	1
17492	25538	EverQuest II	NaN	cool game. THe only thing that REALLY PISSES M...	1
17493	25539	EverQuest II	NaN	this game since I was a little kid, always hav...	1

17494 rows × 5 columns

The user_review column is the plain-text review, and the target which we are trying to predict is user_suggestion, which is a binary value of 0 or 1, indicating whether or not the reviewer ultimately recommended the game to other gamers on Steam. The dataset is well-balanced, with ~7,500 instances of the minority class (not recommended) and ~10,000 instances of the majority class (would recommend).

Of course, the review and user suggestions are both provided by the reviewer, so it will be interesting if some mostly positive reviews don't result in positive suggestions and vice versa.

The data may be split differently for different experiments, but in general, we'll employ a simple random split of 80% train, 10% validation, and 10% test as there is no temporal relevance to consider, nor is stratification necessary.

Game overview information for the train is available in single file game_overview.csv

Title	Developer	Publisher	tags	Overview
Title of the game	Name of the developer of the game	Name of the Publisher of the game	Popular user-defined tags for the game	Overview of the game provided by the publisher

Models:

We will use different classification models to compare the performance and choose the best one. The models includes:

- Auto ML text classification
- Vertex AI text classification

- LogisticRegression
- KNN
- RandomForest
- Neural network(word2vec)

Evaluation Metrics

Sentiment analysis is a classification problem. We will be approaching it as a supervised learning task, as our data is fully labeled. As the target column is binary, we can easily use metrics that are typically used for binary classifiers.

For each model we produce, we'll calculate the following metrics to make for a fair comparison at the end:

- Precision
- Recall
- Accuracy
- F1-score
- AUC/ROC curve

Most approaches to sentiment analysis, such as cloud-based solutions and even the nltk library, offer a multi-class prediction along a spectrum including negative/neutral/positive. For experiments which leverage those libraries, we will gain insights into how mapping a multi-class classifier onto a binary space performs, and what happens when multiple classes are distributed differently into each binary class. For example, we can see what happens if we classify “neutral” as a positive recommendation versus a negative recommendation, and how that impacts our overall metrics.

Programming Language

We will use Python as the programming language, and JavaScript to create the UI dashboard.

Cloud Environment

We will be using Google Cloud to host our model and application for deployment to the end user.

Milestones

Week 9 Oct 28:

- Project selection

Week 10 Nov 4:

- Make the work plan and assignment
- Complete the project proposal
- Create project presentation from template
- Set up new google cloud project
- Extract the data from dataset and transfer to google cloud
- Project Management using the Trello tool
- Create AutoML text classification Import JSON File
- Create Vertex AI text classification dataset from import file in cloud storage
- Add dataset to GCP

Week 11 Nov 11:

- Preprocessing and cleaning the data
- Visualization the data and EDA
- Chi-square Analysis

Week 12 Nov 18:

- Deploy the time-series model and training the data
- Create the dashboard
- Create the pipeline
- Testing the System
- Complete the final report

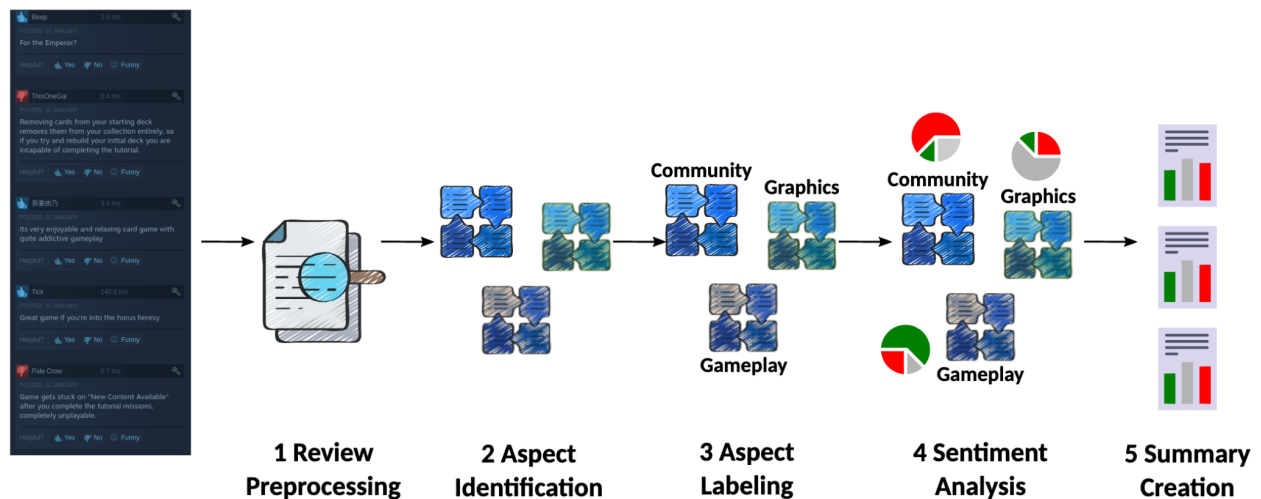
Week 13 Nov 25:

- Prepare the presentation



Architecture of the system

Pipelines

The main components of our pipelines are shown in the figure below.



Detailed Description of our Pipeline

Python	Environmnet	Process Data	Derive Models	Analyze Results	Visualization
	<ul style="list-style-type: none">- Jupiter notebook version 3.6 installed on EC2M5ad.12x large used to import/parse JSON Data	<ul style="list-style-type: none">Create user-item matrix.Pre-process data joining user-item data to game metadata	<ul style="list-style-type: none">Find optimal hyperparameter using 5-fold cross validation.Train final model	<ul style="list-style-type: none">Perform post-traing analysis of model on:<ul style="list-style-type: none">Cold start problemGame bought vs playedBPR vs WARPHybrid model	

References

[1] Sentiment Analysis for Steam Reviews

<https://www.kaggle.com/datasets/piyushagni5/sentiment-analysis-for-steam-reviews>

[2] BigQuery ML on Google Cloud

<https://cloud.google.com/blog/topics/developers-practitioners/how-build-demand-forecasting-models-bigquery-ml>

[3] Machine Learning – Training, Validation & Test Data Set

<https://vitalflux.com/machine-learning-training-validation-test-data-set/>

[4] <https://github.com/cardiffnlp/tweeteval/tree/main/datasets/sentiment>