

# Machine Learning Project Milestone 2: Pitch - Gremlins

Mathew van den Heever & Tyler Walker

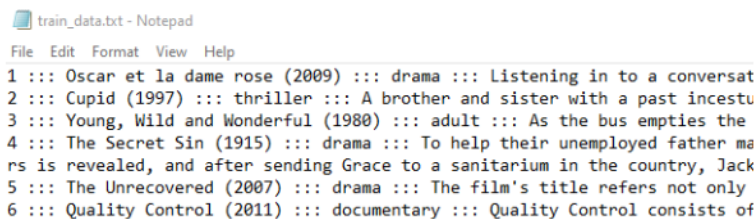
2/11/2022

## I. Problem Space

We considered a wide range of problem spaces when deciding on a project for this class. The first space we considered exploring was development of Angular Distribution Models (ADMs) using NASA CERES data. We decided this problem space was a little too complex, as a group, we would need to spend too much of our time and effort trying to make sense of the data. We then explored using data from poker games to try to quantify if a hand was a strong, weak or moderate hand based on the actions of other players. This data set contained information on many rounds of poker. Most of the information included within the data set was how much each player bet, and where they were seated, but it lacked a lot of critical information that goes into making a hand a good or bad such as how long it took each player to make their bet, and in what fashion the bet was made. We decided we wanted to work with a slightly more simple problem space in order to focus more of our efforts on developing a complex model rather than needing to spend too much time trying to understand and modify our data sets in order to use them in a more simplified model. We found that it would be more interesting to work with text classification. We considered many different problem spaces such as sentiment analysis for movie reviews, which would be binary text classification, but we learned that binary text classification was not going to be challenging enough. We found that multi-class text classification would be more challenging, and we found a good data set for movie descriptions. We will build a model that will classify the genre of a movie based primarily on the movie description.

## II. Data-set

We found the data-set we are planning to use, "Genre Classification Dataset IMDb", on kaggle. The data-set has 108,414 entries, which are already split into two sub-sets, a training set and a testing set. The given training set contains 54,214 entries, and the given testing set contains 54,200 entries. This data-set also has a test solution containing the genres of the test entries. We are considering combining the training set and the testing set to make a set of data which we can split ourselves, using an 80/20 split, with 80% of the data being our training set and 20% of the data becoming our testing set. Making our own split would allow us to have a larger training set to work with, which may help us improve our accuracy. Each entry of the training and testing solution sets have four features. The first is simply an ID that relates each movie to where it occurs in our data-set. The second feature is the title of the movie, with many of the titles and also including the year the movie was released. The third feature is the genre of the movie, which is what we will be predicting. The final feature is a written description of the movie. These four features can be seen in the figure below which is an image sampling the testing data set.



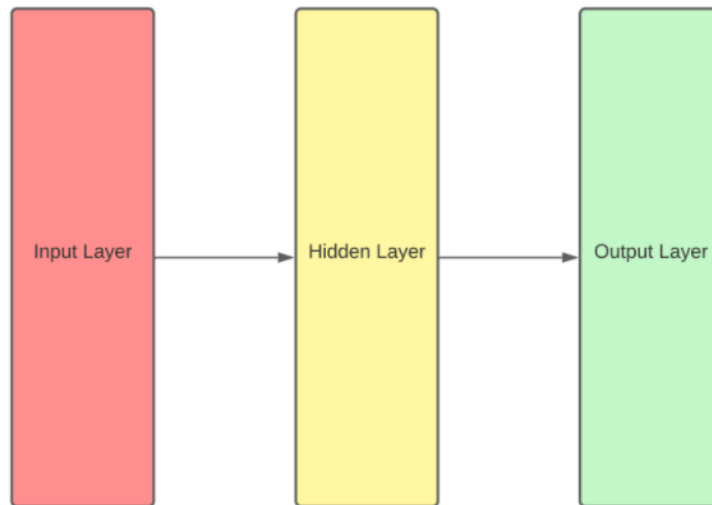
```
train_data.txt - Notepad
File Edit Format View Help
1 ::: Oscar et la dame rose (2009) ::: drama ::: Listening in to a conversat
2 ::: Cupid (1997) ::: thriller ::: A brother and sister with a past incestu
3 ::: Young, Wild and Wonderful (1980) ::: adult ::: As the bus empties the
4 ::: The Secret Sin (1915) ::: drama ::: To help their unemployed father ma
rs is revealed, and after sending Grace to a sanitarium in the country, Jack
5 ::: The Unrecovered (2007) ::: drama ::: The film's title refers not only
6 ::: Quality Control (2011) ::: documentary ::: Quality Control consists of
```

We will need to research methods for pre-processing our data in order to convert the written description of each film into feature vectors. We most likely export these feature vectors to an excel document in order to utilize them in our models. In order to obtain better results we will need to ensure that the our training data set is non-skewed. If the majority of our training data set falls into a single genre, then our model may not be able to, and most likely will not, classify the majority of our entry's genres correctly. Skewness in our training data set can impair the models interpretation of the importance of certain features in our feature vectors.

### III. Planned Approach

We would like to attempt two or three models. The first model is a Naive Bays model and we plan to use the accuracy obtained from this model as our baseline accuracy. Python provides a nice 'out of the box' libraries such as scikit-learn or NLTK to build the Naive Bayes model, which should be straight forward for us. There are many tutorials online that build basic Naive Bayes models using pre-built tools in Python. We anticipate that this will be easy to implement and hopefully we can significantly improve the accuracy with our neural network model or fine-tuned BERT model. In "Naive Bayes and Text Classification I: Introduction and Theory", Sebastian states that "Naive Bayes classifiers are linear classifiers that are known for being simple yet very efficient." We can also analyze what we've learned from building a Naive Bayes classifier and how it compares to our other models. Ideally there will be significant improvement in our other models because they will be much more difficult to implement.

The second model we will attempt will be a neural network model. We are still researching neural network models in an attempt to maximize the accuracy in predicting a films genre. There are many different types of neural network models to consider, below is a figure we made illustrating a basic neural network model.



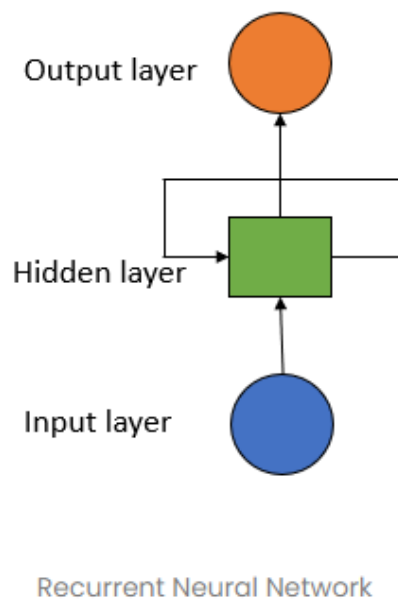
The first type of neural network model that we are looking into would be the recurrent neural network model (RNN). In an article titled "Text Classification with RNN" by Aarya Brahmane, Mr. Brahmane talks about two steps that should be followed "before passing the data into a neural network: embedding and padding". We would like to follow the procedures he uses for embedding and padding in order to pre-process our own data. This will allow us to translate the words in our movie descriptions into a "vector space", as described Mr. Brahmane in his article. The image below is from "Text Classification with RNN" and shows an example of embedding and padding a sentence.

```
sentence=['Fast cars are good',
          'Football is a famous sport',
          'Be happy Be positive']
```

After padding:

```
[[364  50  95 313   0   0   0   0   0   0]
 [527 723 350 333 722   0   0   0   0   0]
 [238 216 238 775   0   0   0   0   0   0]]
```

In building the RNN model, Mr. Brahmane describes that there are three stages. The first stage "moves forward through the hidden layer and makes a prediction". The second stage "compares its prediction with the true value using the loss function". The final stage "uses the error values in back-propagation, which further calculates the gradient for each point (node)." We will need to learn a lot about how to implement this model in python. Brahmane describes using a python library called Tensorflow. Below is an image shown in "Text Classification with RNN", which illustrates a RNN model.



Alternatively, we are considering on looking into a convolution neural network model (CNN). In an article titled "A Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model for Text Classification", Jin Zheng states that "CNN is able to extract local and deep features from natural language and has achieved good results in sentence classification." Zheng states that CNN and RNN have "shown different capability in representing a piece of text". CNN in particular has been "proven to be able to learn local features from words or phrases". Our goal is to find the model that will be the most accurate given our problem space of predicting the genre of a movie. However, we can

imagine that implementing these in python will be challenging and we will settle for the neural network model that is more plausible given our timeline and resources.

Lastly, we are considering implementing a fine-tuned BERT model. Ideally we will be able to build this model in addition to our neural network model. However, we can also get stuck with building the neural network model and decide to work on the fine-tuned BERT model instead. It seems that implementing a fine-tuned BERT model for multi-class text classification would be easier than building a neural network model, but we are not sure. If we are able to implement all a Naive Bayes model, neural network model, and a fine-tuned BERT model, it will be interesting to analyze the differences between the three.

Multi-class text classification is not a particularly novel idea nor is genre classification for movies. We have decided to work in a fairly simple project space with the hopes that we are able to spend our time learning and understanding how these machine learning models can be created, how they work and how to improve the ones models in order to obtain the best results. We are also going to focus some of our time on learning how to evaluate the effectiveness and accuracies of our models which will be helpful when we start trying to applying these concepts to future projects.

#### **IV. Stretch Goals**

We have a few stretch goals in mind for this project. The first stretch goal is to beat our baseline accuracy via a neural network. Our second stretch goal is to build both a neural network model and a fine-tune BERT model if time permits. If we are able to implement both of these models, then we will also attempt to compare their accuracies, not only with the Naive Bays model but also with each other to see which is the most effective model. Our final stretch goal is to try to determine when the film was released. As mentioned in the data-set section above, many of the entries contain the year the film was released in the same feature as the title of the film. If we have time to first implement all three models, then we will start trying to be able to determine when the movie was released based on the language used in the description. Initially we would try to build a model that can determine the correct decade and if we have time, we are hoping to improve upon this and try to determine the exact year the movie was released. One possible extra feature that we may end up using for this stretch goal is the movies genre. It will be interesting to see if certain genres have a better correlation to certain times, such as horror movies becoming more and more popular in recent years.

#### **V. Citations**

- Raschka, Sebastian. "Naive Bayes and Text Classification I - Introduction and Theory." ArXiv.org, 14 Feb. 2017, <https://arxiv.org/abs/1410.5329>.
- Team, Towards AI Editorial. "Text Classification with RNN." Towards AI - The World's Leading AI and Technology Publication, Towards AI, 21 Nov. 2020, <https://towardsai.net/p/deep-learning/text-classification-with-rnn>.
- "A Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model for Text Classification." IEEE Xplore, IEEE, 1 Aug. 2019, <https://ieeexplore.ieee.org/abstract/document/8784247>.