Machine Learning & Data Mining
Miniproject 1

Caltech CS/CNS/EE 155
Released January 28th, 2017

Introduction

- **Group members**
  Vaibhav Anand
  Nikhil Gupta
  Michael Hashe

- **Team name**
  The Breakfast Club

- **GitHub link**
  https://github.com/nkgupta1/CS155-Project1-Voter

- **Division of labour**

Overview

- **Models and techniques tried**
  - **Bullet:** Bullet text.

- **Work timeline**
  - **Bullet:** Bullet text.

Approach

- **Data processing and manipulation**
  - **Original Data:** At first, we trained the models on the raw data. These models ended up performing relatively poorly on the leaderboard. In looking closer at the data, we realized this was because of the way the data was labeled: questions with categorical answers had integers as labels for the different categories. As such, the models were trying to put an ordering to the dimensions where none existed, which hurt the accuracy of the models.
  - **All Categories:** The first thing we tried in order to resolve this issue was to flatten the whole data set and make it all categorical. In particular, this choice was motived by a cursory look through the handbook, from which we noticed that a majority of the dimensions were categorical. Before we invested time in splitting the data into ordered and unordered dimensions, we tried training models when the whole data was made into categories. Models trained on this data had memory issues because of the size of the data set and there a very little chance of generalization because of the number of categories present.

- **Categories and Ordered:** After taking a closer look through the handbook, we realized that many of the issues we were having with categorizing the data was due to the specific characteristics of the dimensions. While most of the dimensions were unordered, many of them still had an ordering. As such, this forced us to manually sift through the data in order to bin the dimensions into ordered and unordered. In order to ease this process, we wrote a parser so that we could later modify which dimension were kept and how they were treated.

  The range of each of the ordered dimensions was mapped to [0,1] so that the models could better deal with them. This choice was made after trying to normalize the data and having some of the values still being very large. The unordered data was made into categories with one category for each of the values present. The parser kept track of the way each dimension was processed so that the test data could be processed in the same way. This raised an issue of how to treat values for dimensions that were in the test set but not in the training set for the unordered dimensions (while this occurred, it was not particularly frequent). We ended up putting no value for this because we thought it would be better to omit data than to lie about the data.

  After putting the data through this processing, the number of resulting dimension was around 50000, which is too many for the number of points that we had. In looking through how many dimensions the processor made each of the original dimensions into, there was one dimension that accounted for all but about 4000 of the dimensions, a unique ID for each family. As such, we added another option to the parser to completely delete dimensions. Models trained on this new, processed data set performed much better than the models performed on the original data set. We used NumPy for much of the data processing because of the number of methods present in the NumPy library which meant that we did not have to reimplement many of the functions. Furthermore, NumPy methods run very quickly on the NumPy arrays.

- **Trimming Dimensions:** After using the last data set for most of the models, we were not able to improve on accuracy much more so we thought this might be due to noise in the dimensions of the data set. As such, we made the decision to go through the handbook more carefully and make the decision to keep or delete each of the dimensions. This is because there were a significant number of dimensions that added noise to the dataset. For example, there were many dimensions for line number which was not something that the people actually answered, so were something that we decided were insignificant. After this final trimming process, the number of dimensions present were reduced from about 4000 to 3000. This reduced dimensionality improved the performance of some of the models.

- **Details of models and techniques**
  - **Bullet:** Bullet text.

Model Selection

- **Scoring**

- **Validation and Test**

Conclusion

- **Discoveries**

- **Challenges**

- **Concluding Remarks**