Team 2 CSC 177 Presentation

Members:
David Castrejon, Hashem Jaber, Shahed Jaber, Fernando, Kevin Cendana, Dip Patel, Albin Shabu

Project #1 Data Preprocessing

Data Set: So much candy data, seriously!

Our team decided to use the “So much candy data, Seriously” dataset. This data sheet provided insights into the candy preferences of individuals along with their personality traits. We analyzed the relation between individuals’ likings and their responses to open ended personality questions.

By preprocessing this dataset, analysis would be able to be performed to answer questions such as:

1. Based on what is someone’s favorite color and reality tv show preference, what candy would they like?

2. What age ranges have trends for candy preferences?

3. Does gender play a role in selection of candy?

4. Does geographical location have a correlation with candy preferences?

Summary:

We focused on applying preprocessing techniques such as:

Concatenation: Using hot encoding, we were able to combine columns to reduce redundancy

Fill Null Values: For the column “Trick or treating?”, we replaced all null values with a “yes” or “no” based off the respective percentage of such answers in all valid rows

Dropping Columns: Some columns allowed for open ended responses with no input validation, allowing for randomized and irrelevant data

Hot Encoding: For the responses for the liking of a particular candy, there were only 4 types of answers which allowed us to hot encode the columns

Removing Outliers: The age column contained responses completely outside the valid life expectancy of a human, and such values were detected and removed during preprocessing

Project #2 Linear Regression Project & Classification Tree

Linear Regression Project & Classification Tree

We analyzed more than one set of data. We had to apply regression on all the data sets provided and plot all the data. We had to use both Simple Linear Regression model and Multiple Regression Model to fit our data. So for the admission data set, we used some features to predict the percentage of someone’s chance of admission to a college based on the features chosen.

Things we focused on:

1. Linear and multilinear regression.
2. Linear and multilinear regression on given dataset
3. Classification on given dataset
4. ID3 entropy problem

Summary:

We focused on applying preprocessing techniques such as:

Hot encoding: One-hot encode the columns of the DataFrame based on the specified threshold and minimum number of unique values and dropped the original columns that were one-hot encoded.

Cleaning: we cleaned the age column to only have number and replaced the non-numbersl with the median of the age column and also converted to numbers rounded to nearest whole number. We also cleaned country and only included unique names and made them lowercased.

Choosing features: we chose what features we wanted and dropped the features that we did not want from our candyhierarchy2017.csv

Multiple regression: we used multiple regression using a neural network. It separates the data into features and target values, builds a neural network model with two hidden layers and one output layer, compiles the model, trains it on the training data while validating its performance on a separate test set

Project #3 Classification Models project

Data Set: Churn Rate Dataset

The Churn Rate Dataset is the percent of subscribers who terminate their subscriptions within a specified timeframe. For a company to broaden its customer base, the classification problem is where the goal is to predict whether an individual will EXIT (1) or NOT (0) based on their Region, Gender, Age, Tenure, Balance, Salary etc

focused on applying the following classification techniques:

1. Naïve Bayes

2. K-Nearest Neighbor (KNN)

3. Support Vector Machines (SVM)

4. Decision Trees (DT)

5. Logistic Regression (Logit)

Summary:

K-Nearest Neighbors (KNN): Optimal at 19 neighbors with 83.95% accuracy, but biased towards predicting non-cancellation of subscriptions.

Decision Trees: Applied to the Titanic dataset, focusing on socio-economic and personal attributes, achieving 77.24% accuracy.

Support Vector Machine (SVM): Predicted subscription exits with 79.35% accuracy, but was heavily biased towards non-exit predictions.

Logistic Regression: Achieved around 79.7% accuracy in predicting subscription exits, but had lower precision and recall, indicating some prediction challenges.

Project #4: Cluster Analysis, ANN & Text Mining Project

Dataset: imdb_dataset.csv

Cluster Analysis: K-Means Clustering

We used K-Means to study a bunch of movie titles from the IMDB dataset.

First, we cleaned up the dataset by getting rid of some columns we didn’t need.

We then used TF-IDF to turn the movie titles into a format that the computer can understand and analyze.

To figure out how many groups (clusters) to divide these titles into, we used the elbow method. This method helps us see at what point adding more clusters doesn’t really help us get better information

We decided that the best number of groups for our data was 3.

Dataset: imdb_dataset.csv-contin

Text Mining

Preprocessing:

We used three different ways (single, complete, average) to see how the titles can be grouped based on how similar they are.

We took a set of sentences and used two techniques: Count Vectorizer and TF-IDF Vectorizer.

Countvectorizer counts how many times each word shows up.

TF-IDF Vectorizer not only counts words, but also checks how unique these words are across all sentences. Basically, it helps us figure out which words are really important in a sentence.

We showed these counts and TF-IDF scores in a table and talked about why TF-IDF is useful. It’s great for finding important words in a bunch of text, which can be handy for things like sorting documents or finding what a web page is about.

Summary:

Cluster Analysis: We used K-means to study a batch of movie titles from the IMDB dataset. We removed any unneeded columns & used TF-IDF to convert it into a readable format.

In order to find out how many groups (clusters) to divide these titles into, we used the elbow method to see at what point adding more clusters doesn’t really help us get better information.

Text Mining: We used a count vectorizer to tally up the words and the TF-IDF vectorizer to check how unique these words were across all sentences. This was done in order to find out which words were most important in the data set. In real life scenarios, this can be handy for sorting through documents or quickly finding out the main subject of a web page.

Kachow!