Deep Learning Algorithms

AMS 573 Categorical Data Analysis

Date: May 1st, 2023

Lieu: Stony Brook University

By: Jamila Awad

Goal of presentation

- Present deep learning tools for categorical data analysis
- Provide deep learning toolbox: encode variables, deep learning regressions and k-means clustering.
- Implement deep learning toolbox with examples: breast cancer dataset, drug reviews dataset
- Present study results
- Provide limitations
- Compare advantages vs disadvantages

Connection deep learning-categorical data

- **Deep learning**: AI branch that uses neural networks to extract various layers of data.

- **Deep learning vs machine learning**: It understand features incrementally, eliminates need of domain expert.

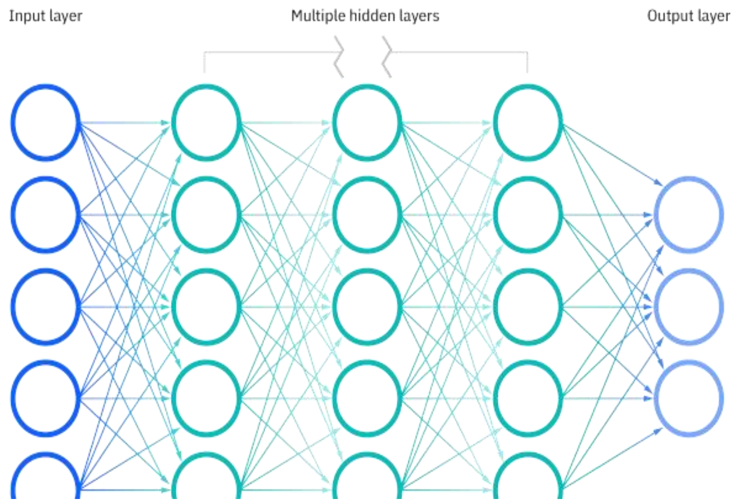
- **Categorical data analysis**: organizes a response variable into a set of mutually exclusive ordered or unordered categories.

- **Applications**: bioinformatics, image recognition, NLP

Neural Networks

- **Main attributes**: Heart of deep learning, inspired by the human brain
- **Configuration**: Comprised of node layers, containing an input, several hidden layers and an output layer. Each node connects to another. When the node is activated, it sends data to the next layer of the network.

How to enhance algorithm accuracy?
Tune hyperparameters



Process time?
From seconds to hours

Image Source: [What are Neural Networks? | IBM]

Neural Networks

- **Various forms**: recurrent neural networks, convolutional neural networks, artificial neural networks, and feedforward neural networks.
- **How do they operate**? They feed data in and let the model figure out for itself whether it has made the right interpretation or decision about a given data element. Trial-and-error process that works with iterations.
- **How it processes categorical data**? It passes it into a neural network that is then used to envision models.

Organigram deep learning toolbox

Encoding variables	Regressions	K-means clustering
• Ordinal encoding • One hot encoding	• Gradient boosting regression • Grid gradient boosting regression • Linear SVM regression • BernoulliNP • Random forest regression • MLP regression	• k-means = 3

Encoding variables

- **2 techniques examined**: ordinal encoding and one hot encoding.
- **Ordinal encoding**: simple form, naïve way of encoding variables, maps each unique label to an integer.
- **One hot encoding**: robust and efficient technique, maps each label to a binary vector where one element to each unique label is encoded by 1 and all other elements are encoded by 0.

Deep learning regressions

RMSLE:

Metric to compare regression performance, the lower the better

1. **Gradient boosting**: Adds decision trees to the next decision tree that corrects the previous decision error
2. **Grid gradient boosting**: Uses grid search to obtain the best set of hyperparameters
3. **Linear SVM**: Uses support vectors to first class data then finds a hyperplane with the maximum number of points
4. **BernoulliNP**: naïve, assumes features conditional independence, uses Bayes theorem
5. **Random Forest**: creates multiple decision trees by splitting the dataset then makes predictions by taking the average of individual trees
6. **MLP Regression**: uses backpropagation to adjust weights between neurons and uses many layers of perceptrons

K-means clustering

- **Objective**: minimize within-cluster variances
- **Cluster**: collection of data points aggregated sharing similarities
- **How does it operate**?

1. Partitions n observations into k clusters
2. Measures distance between centroid (=cluster with the nearest mean). Centroid serves as a prototype to the cluster
3. Performs iterative calculations to optimize position of centroids

Data Coding

General packages Begin →	`#load packages`
	`from pandas import read_csv`
	`import pandas as pd`
	`import numpy as np`
	`import matplotlib.pyplot as plt`
	`import matplotlib.dates as mdates`
	`from pandas import DataFrame`
General packages End →	`import seaborn as sn`
	`#deep learning packages`
	`from sklearn.model_selection import train_test_split`
	`from sklearn.preprocessing import LabelEncoder`
Encode categorical variables →	`from sklearn.preprocessing import OrdinalEncoder`
	`from sklearn.preprocessing import OneHotEncoder`
Deep learning regressions Begin→	`from keras.models import Sequential`
	`from keras.layers import Dense`
	`from sklearn.ensemble import GradientBoostingRegressor`
	`from sklearn.model_selection import GridSearchCV`
	`from sklearn.svm import SVC`
	`from sklearn.svm import LinearSVC`
	`from sklearn.naive_bayes import BernoulliNB`
	`from sklearn.linear_model import LinearRegression`
	`from sklearn.linear_model import LogisticRegression`
	`from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier`
	`from sklearn.metrics import classification_report`
	`from sklearn.metrics import accuracy_score`
	`from sklearn.neural_network import MLPClassifier`
Deep learning regressions End→	`from sklearn.utils import shuffle`
K-means clustering→	`from sklearn.datasets import make_blobs`
	`#load sentiment package`
	`from textblob import TextBlob`
	`#computational system`
CPU time→	`import time`

Case study

- **Data source**: UCI machine learning repository

- **Objective**: Compare 2 opposing datasets

1. Breast Cancer: medium size clean “picture perfect“ dataset
2. Drug Reviews: messy noisy unperfect massive dataset

Breast cancer:

UCI Machine Learning Repository: Breast Cancer Data Set

Drug reviews:

UCI Machine Learning Repository: Drug Review Dataset (Drugs.com) Data Set

Implementation structure

- **Encoding variables**: Split columns into input (X) and output (Y), reshape output, use train_test split ratio 2:1, then use encode scikit-learn packages. It maps 2 class labels in a neural network containing 1 hidden layer with 10 nodes. Measure accuracy and CPU time.
- **Regression**: Use train_test split ratio 2:1 then use encode scikit-learn and Keras packages. Code RMSLE functions. Measure RMSLE and CPU time.
- **K-means clustering**: Set k=3 clusters, use make_blobs function to cluster dataframe, visualize raw data, code initial centroid function to measure distance btwn centroid & cluster, assign cluster to class labels then measure change of distance in centroids. Plot clusters.

Results breast cancer dataset

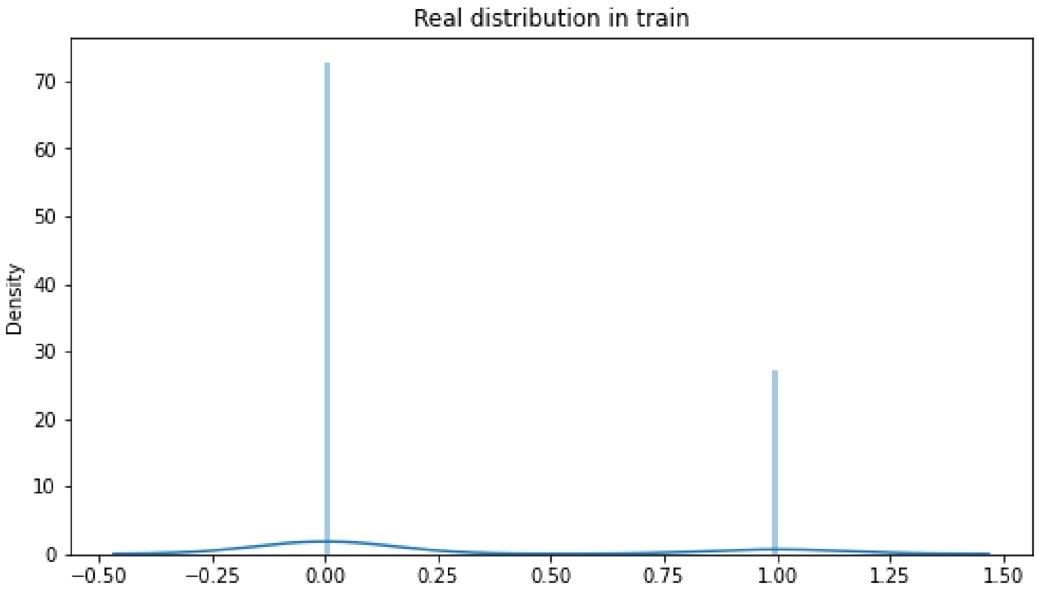
Table 1: Categorical Variables Encoder Breast Cancer

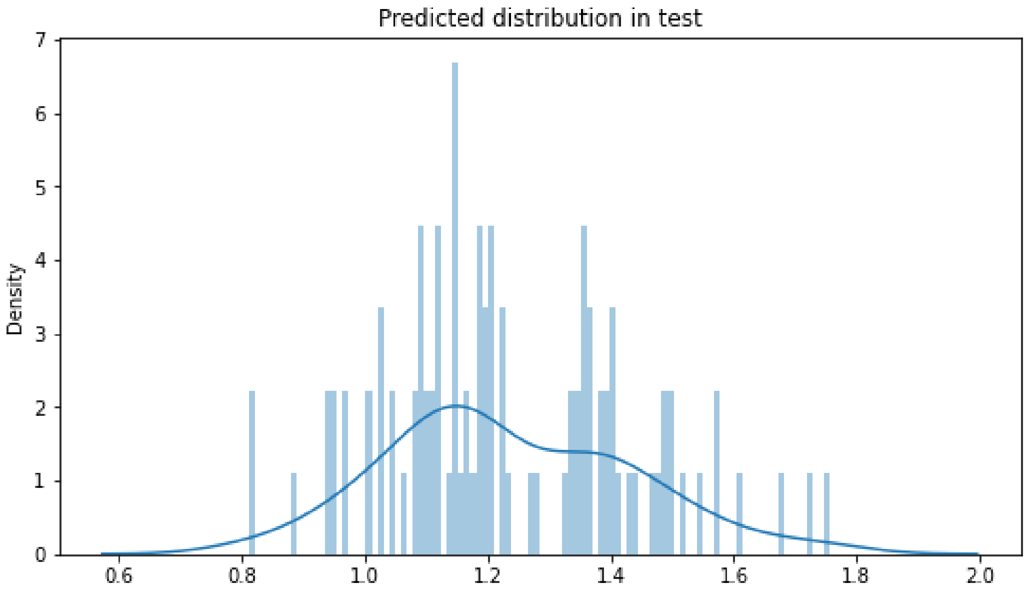
Method	Accuracy	CPU execution time (seconds)
Ordinal encoder	68.42	4.2969
One hot encoder	72.63	4.2188

Winner: One hot encoder with 72.63% accuracy and fastest CPU time at 4.2188 seconds.

Results breast cancer dataset

<u>Figure 1: Train and Test Distribution Simple Linear Regression</u>





<u>Observation</u>: Distribution between train and test varies significantly → not adequate for categorical variable modeling.

Results breast cancer dataset

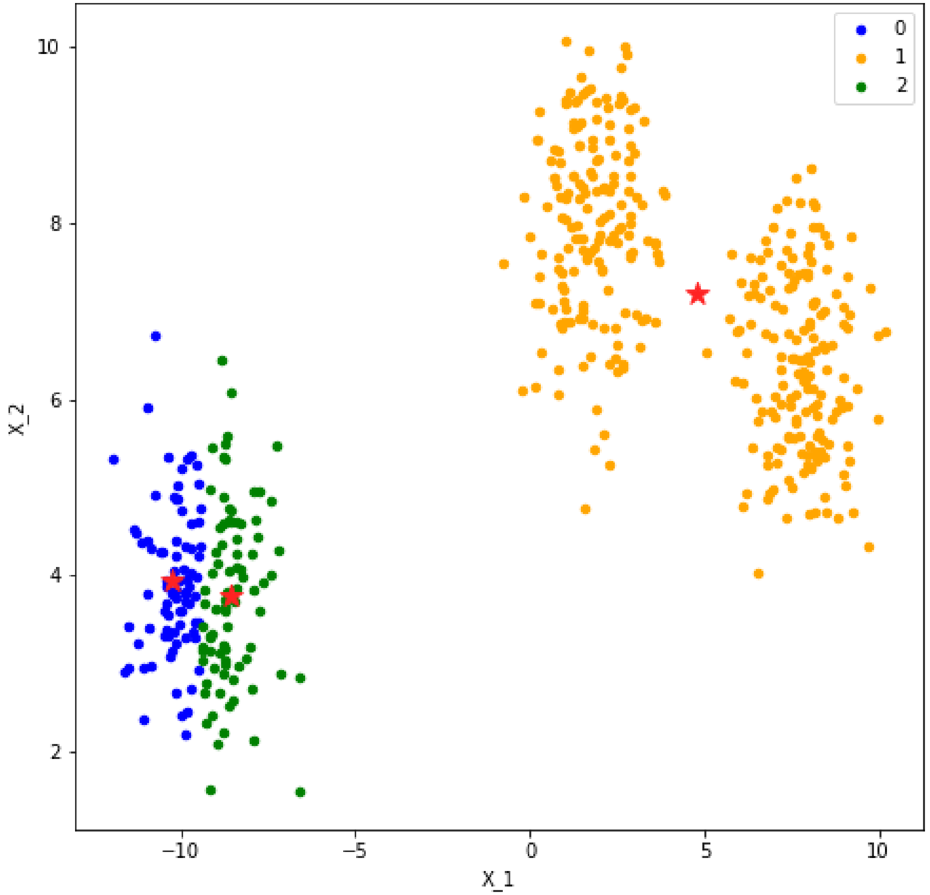
Table 2: Deep Learning Regression Performance

Method	RMSLE	CPU execution time (seconds)
Simple linear regression	0.64496	0.01563
Gradient boosting regression	0.65146	0.07815
Grid gradient boosting regression	0.66252	26.96875
Linear SVC regression	0.67368	0.00002
BernoulliNP	0.75789	0.01562
Random forest regression	0.71579	0.04687
MLP regression	0.66315	0.03125

Winner: Gradient boost RMSLE 0.65146 but in detriment to CPU time 0.07815 seconds

Results breast cancer dataset

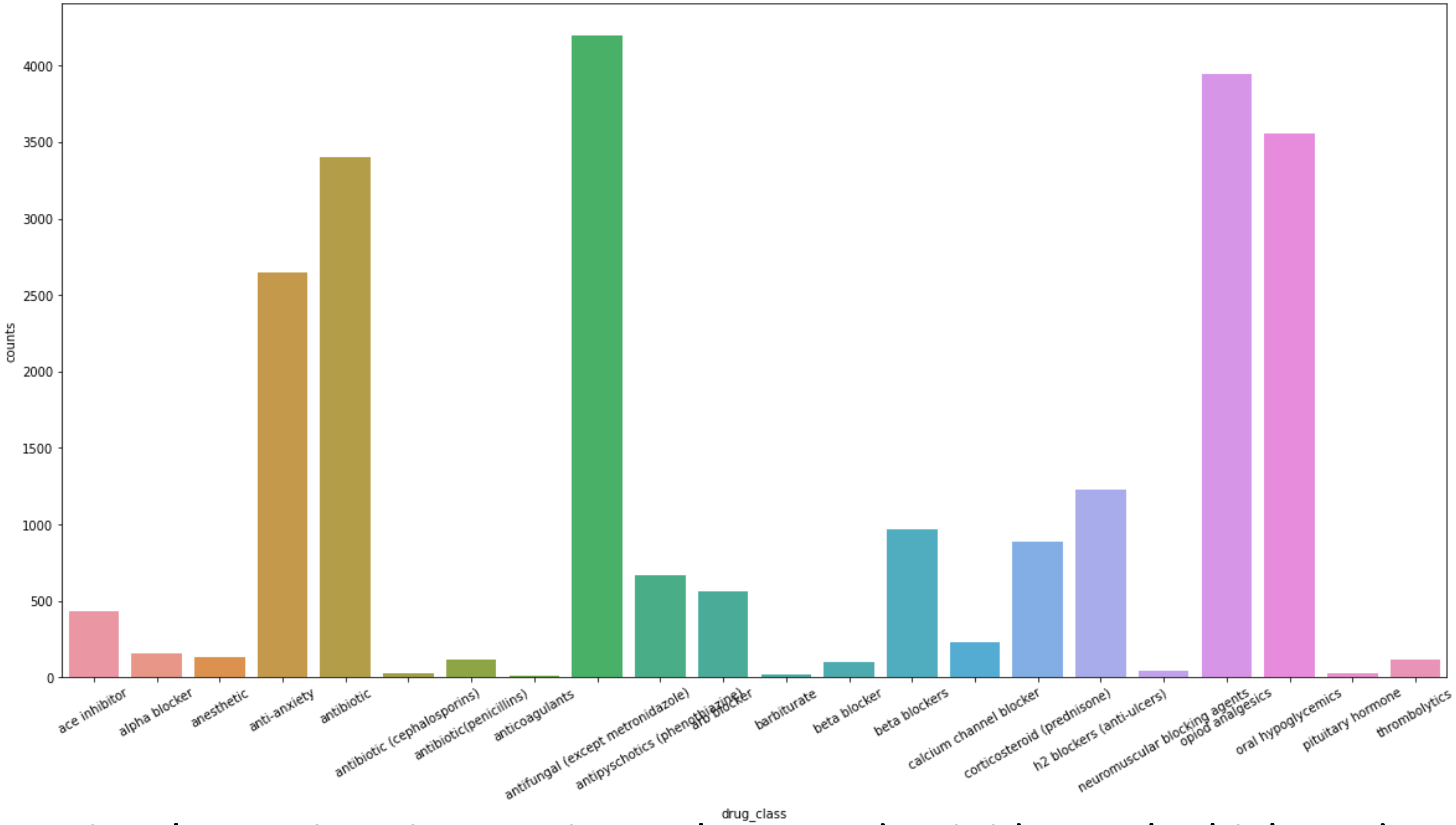
<u>Figure 2: K-means Clustering Algorithm</u>



<u>Observation</u>: 3 clusters with centered red star centroids, CPU time 4.26563 seconds → good model for this dataset

Results Drug Reviews

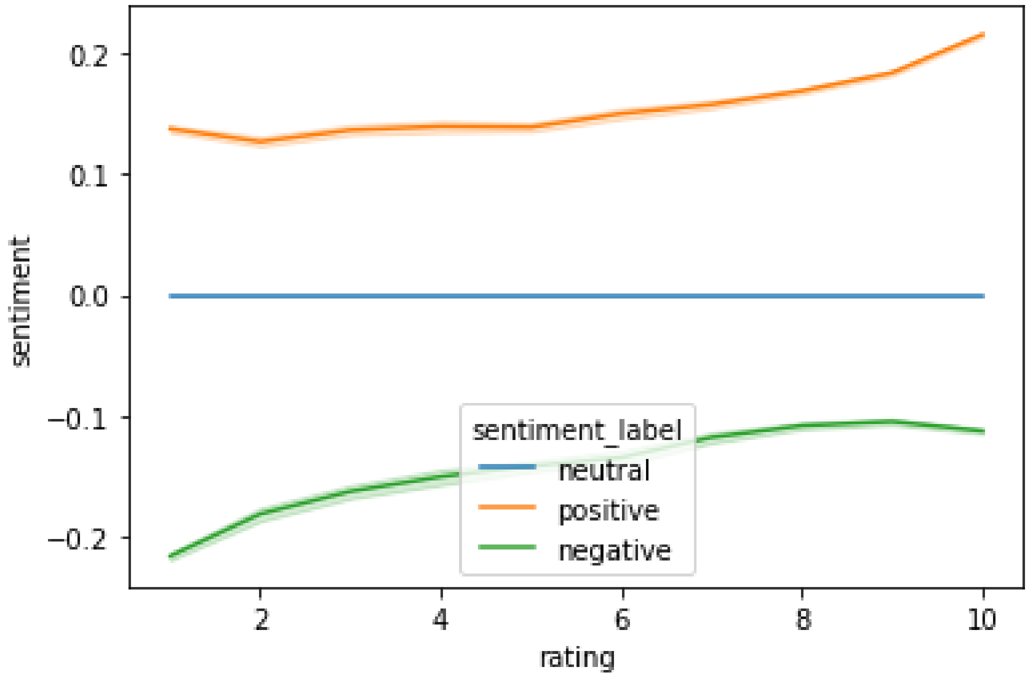
Figure 3: Drug Groups by Counts



Observation: We notice that anti-anxiety, anticoagulants, and opioids are the highest drug groups by counts.

Results Drug Reviews

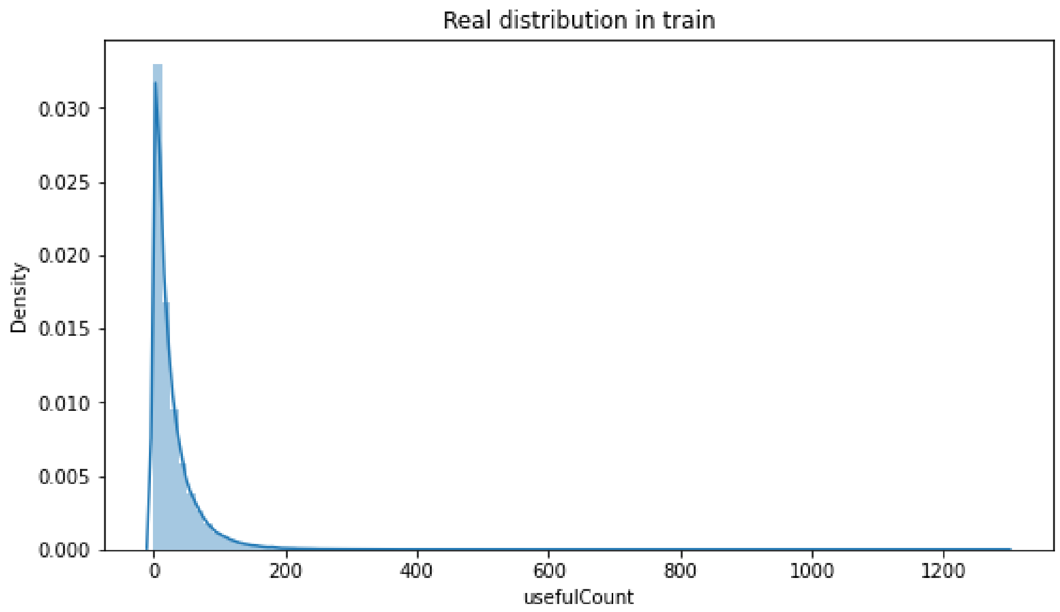
Figure 4: Sentiment Reviews for Drug Ratings

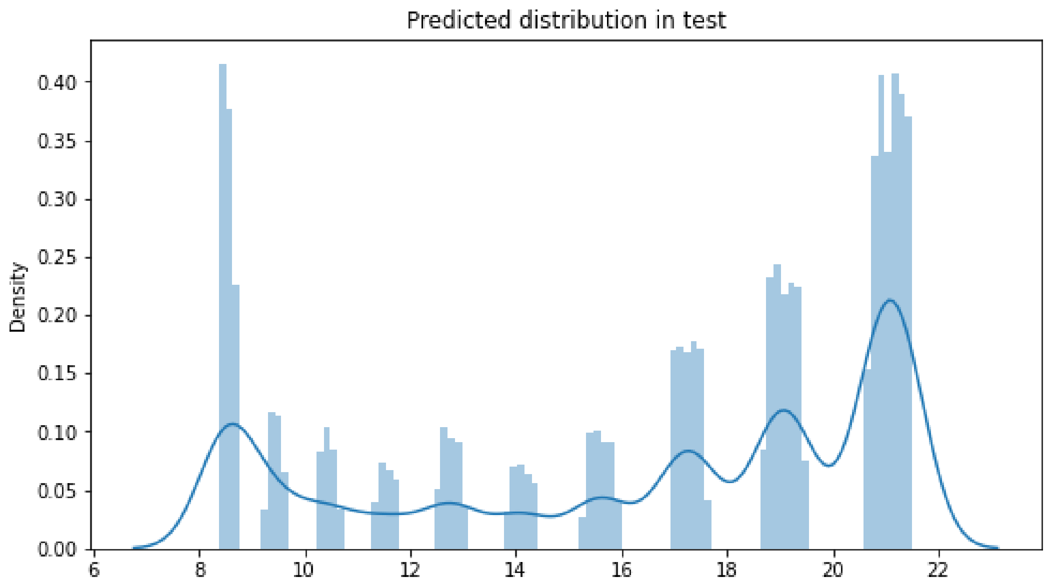


Observation: Positive reviews → good sentiment & negative reviews → bad sentiment

Results Drug Reviews

Figure 5: Train and Test Distribution Simple Linear Regression





Observation: Distribution between train and test varies significantly → not adequate for categorical variable modeling.

Results Drug Reviews

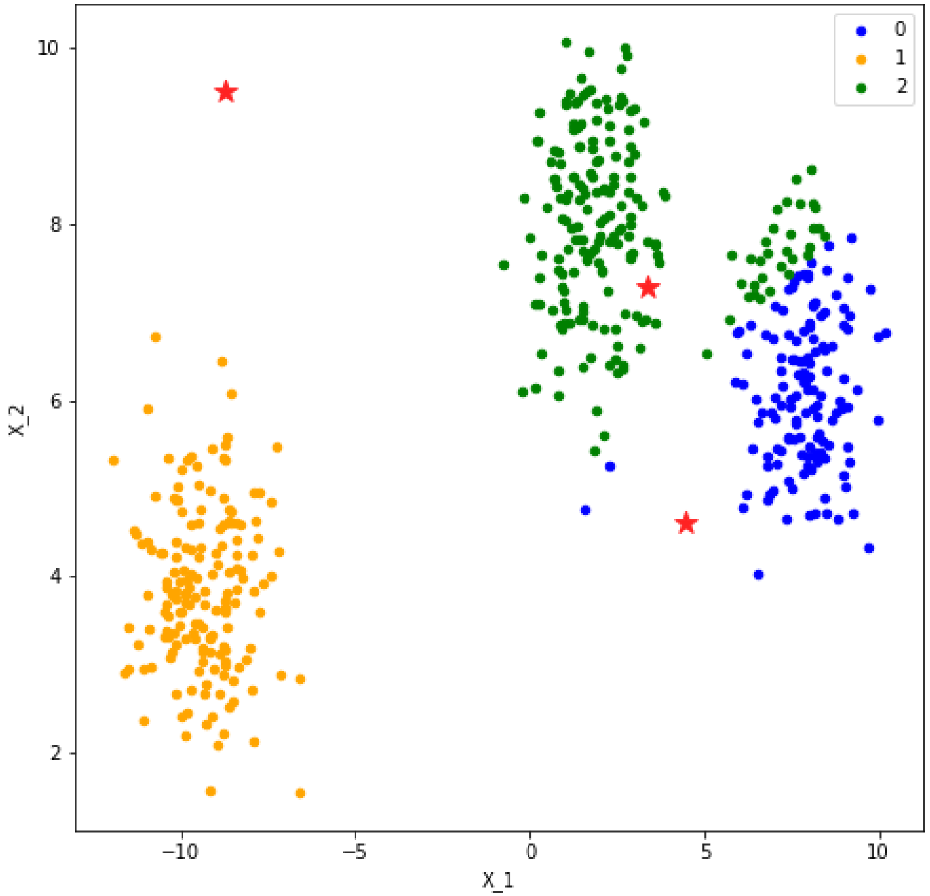
Table 3: Deep Learning Regression Performance

Method	RMSLE	CPU execution time (seconds)
Simple linear regression	1.14837	0.37500
Gradient boosting regression	1.07205	7.45312
Grid gradient boosting regression	0.87036	3154.60937
Linear SVM regression	0.00028	1628.40625
BernoulliNP	0.03897	2.04687
Random forest regression	0.91111	0.06250
MLP regression	0.03864	102.37500

Winner: Linear SVM RMSLE 0.00028 but in detriment to CPU time 1628.40625 seconds

Results Drug Reviews

<u>Figure 6: K-means Clustering Algorithm</u>



<u>Observation</u>: 3 clusters with non-centered red star centroids, CPU time 0.31250 seconds → bad model for this dataset

Comparison Breast Cancer vs Drug Reviews

Breast Cancer

✓One hot encoding

✓Gradient boost regression

✓K-means clustering

Drug Reviews

X Encoding variables

✓ SVM linear regression

X K-means clustering

Conclusion: No perfect universal fit in deep learning algorithms to model categorical variables

Limitations

- Deep learning algorithm downside: It learns through observations from a trained dataset, not practical for sparse data, if trained data has bias then it produces flawed model.

- Deep learning algorithms: All based on hyperparameters (i.e. learning rate) that need to be tuned.

- Variable encoding: ordinal has difficulty distinguishing 2 categories with same frequency, one hot tends to overfit attributes that contain unique values.

Limitations Regressions

1. **Gradient boosting**: use trees sequentially vs CPU time
2. **Grid gradient boosting**: grid search & sequential trees vs CPU time
3. **Linear SVM**: pick right kernel to perform massive datasets
4. **BernoulliNP**: too simplistic to tune parameters
5. **Random Forest**: number of trees vs CPU time
6. **MLP Regression**: quality of training of multiple layer perceptrons vs CPU time

Limitations k-means clustering

1. Requires to specify number of clusters (K) in advance
2. Not suitable to identify clusters with non-convex shape
3. Does not handle noisy data and outliers (example: drug reviews dataset)

Conclusion

- Deep learning tools: powerful systems to process large scale datasets and provide high-performance graphics.

- Drawback: Inflexible one trained and better suited to solve a specific problem

- Categorical data analysis: Deep learning provides better faster models for feature introspection without supervision.

References

- [1] Charis J., Exploratory Data Analysis in Python using the Drug Review Dataset (2020).
- https://github.com/Jcharis/data-science-projects/tree/master/exploratory_data_analysis_in_python_drug_reviews_dataset
- [2] Hancock J. and Khoshgoftaar T., Survey on categorical data for neural networks, Journal of Big Data (2020) 7:28.
- Survey on categorical data for neural networks | Journal of Big Data | Full Text (springeropen.com)
- [3] Hayashi Yoichi, Does Deep Learning Work Well for Categorical Datasets with Mainly Nominal Attributes? (2020), Meiji University
- Does_Deep_Learning_Work_Well_for_Categorical_Datas.pdf
- [4] Jankovic R. and Amelio A., Comparing Multilayer Perceptron and Multiple Regression Model for Predicting Energy Use in the Balkans (2018)
- 1810.11333.pdf (arxiv.org)
- [5] Li Y. and Wu H., A Clustering Method Based on K-Means Algorithm (2012) 25, SciVerse ScienceDirect.
- (PDF) A Clustering Method Based on K-Means Algorithm (researchgate.net)
- [6] Potdar K. et al., A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers, International Journal of Computer Applications (2017) 4:175.
- (PDF) A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers (researchgate.net)
- [7] Schonlau M and Zou R., The random forest algorithm for statistical learning (2020) 1:20, The Stata Journal.
- The random forest algorithm for statistical learning (sagepub.com)
- [8] Sukanaya V. and Seby J., A Machine learning approach for Industry classification Using resume data, IJCRT (2021) 9:10.
- IJRTI (ijcrt.org)
- [9] Zemel R. and Pitassi T., A Gradient-Based Boosting Algorithm for Regression Problems. University of Toronto.
- A Gradient-Based Boosting Algorithm for Regression Problems (neurips.cc)

Questions?

Image source: [Question Time | The Pukeko Patch]

The End

Thanks for being a great audience!