Data User Demographics

Josh Lim

Agenda

1. Data Introduction
2. Exploratory Data Analysis/Data Wrangling
3. Importance of Data
4. Classification Model Analysis
5. Discuss Dimensionality Reduction Process
6. Cluster Analysis
7. Neural Network Classification Analysis
8. Limitations
9. Places for Improvements

About the Data

- Data was obtained from a Kaggle Competition
- Six CSV files each containing information on cell phone data
    - Age/Gender of User
    - App Information
    - Device Information
    - Etc.
- Each CSV consisted of around 32 Million rows

Exploring the Data and Data Wrangling

```
app_events.head()
```

	event_id	app_id	is_installed	is_active
0	2	5927333115845830913	1	1
1	2	-5720078949152207372	1	0
2	2	-1633887856876571208	1	0
3	2	-653184325010919369	1	1
4	2	8693964245073640147	1	1

```
events.head()
```

	event_id	device_id	timestamp	longitude	latitude
0	1	29182687948017175	2016-05-01 00:55:25	121.38	31.24
1	2	-6401643145415154744	2016-05-01 00:54:12	103.65	30.97
2	3	-4833982096941402721	2016-05-01 00:08:05	106.60	29.70
3	4	-6815121365017318426	2016-05-01 00:06:40	104.27	23.28
4	5	-5373797595892518570	2016-05-01 00:07:18	115.88	28.66

```
phone_brand.head()
```

	device_id	phone_brand	device_model
0	-8890648629457979026	小米	红米
1	1277779817574759137	小米	MI 2
2	5137427614288105724	三星	Galaxy S4
3	3669464369358936369	SUGAR	时尚手机
4	-5019277647504317457	三星	Galaxy Note 2

```
gender_age_train.head()
```

	device_id	gender	age	group
0	-8076087639492063270	M	35	M32-38
1	-2897161552818060146	M	35	M32-38
2	-8260683887967679142	M	35	M32-38
3	-4938849341048082022	M	30	M29-31
4	245133531816851882	M	30	M29-31

```
category_labels.head()
```

	label_id	category
0	1	NaN
1	2	game-game type
2	3	game-Game themes
3	4	game-Art Style
4	5	game-Leisure time

```
app_labels.head()
```

	app_id	label_id
0	7324884708820027918	251
1	-4494216993218550286	251
2	6058196446775239644	406
3	6058196446775239644	407
4	8694625920731541625	406

	event_id	app_id	is_installed	is_active	device_id	timestamp	longitude	latitude	label_id	category	phone_brand	device_model	gender	age	group
0	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	549	Property Industry 1.0	华为	Mate 7	M	19	M22-
1	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	548	Industry tag	华为	Mate 7	M	19	M22-
2	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	710	Relatives 1	华为	Mate 7	M	19	M22-
3	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	704	Property Industry 2.0	华为	Mate 7	M	19	M22-
4	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	172	IM	华为	Mate 7	M	19	M22-

	event_id	app_id	is_installed	is_active	device_id	timestamp	longitude	latitude	label_id	category	phone_brand	device_model	gender	age	group
0	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	549	Property Industry 1.0	华为	Mate 7	M	19	M22-
1	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	548	Industry tag	华为	Mate 7	M	19	M22-
2	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	710	Relatives 1	华为	Mate 7	M	19	M22-
3	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	704	Property Industry 2.0	华为	Mate 7	M	19	M22-
4	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	172	IM	华为	Mate 7	M	19	M22-

- 三星 samsung
- 天语 Ktouch
- 海信 hisense
- 联想 lenovo
- 欧比 obi
- 爱派尔 ipair
- 努比亚 nubia
- 优米 youmi
- 朵唯 dowe
- 黑米 heymi
- 锤子 hammer
- 酷比魔方 koobee
- 美图 meitu
- 尼比鲁 nibilu
- 一加 oneplus
- 优购 yougo
- 诺基亚 nokia
- 糖葫芦 candy
- 中国移动 ccmc
- 语信 yuxin
- 基伍 kiwu
- 青橙 greeno
- 华硕 asus
- 夏新 panosonic

- 维图 weitu
- 艾优尼 aiyouni
- 摩托罗拉 moto
- 乡米 xiangmi
- 米奇 micky
- 大可乐 bigcola
- 沃普丰 wpf
- 神舟 hasse
- 摩乐 mole
- 飞秒 fs
- 米歌 mige
- 富可视 fks
- 德赛 desci
- 梦米 mengmi
- 乐视 lshi
- 小杨树 smallt
- 纽曼 newman
- 邦华 banghua
- E派 epai
- 易派 epai
- 普耐尔 pner
- 欧新 ouxin
- 西米 ximi
- 海尔 haier
- 波导 bodao
- 糯米 nuomi

- 唯米 weimi
- 酷珀 kupo
- 谷歌 google
- 昂达 ada
- 聆韵 lingyun

```
"华为": "huawei",  # manually translated and entered
"小米": "xiaomi",  # manually translated and entered
"魅族": "meizu",  # manually translated and entered
"vivo": "vivo",  # manually translated and entered
"酷派": "coolpad",  # manually translated and entered
"索尼": "sony",  # manually translated and entered
"OPPO": "oppo",  # manually translated and entered
"LG": "lg",  # manually translated and entered
"HTC": "htc",  # manually translated and entered
"金立": "gionee",  # manually translated and entered
"中兴": "zte",  # manually translated and entered
"奇酷": "qiku",  # manually translated and entered
"TCL": "tcl",  # manually translated and entered
```

Removing Unnecessary Columns

- After examining the new merged and converted dataframe, these columns did not seem like they would be contributing to the analysis
- Columns: "is_installed", "timestamp", "latitude", "longitude", "category"

	event_id	app_id	is_installed	is_active	device_id	timestamp	longitude	latitude	label_id	category	phone_brand	device_model	gender	age	group
0	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	549	Property Industry 1.0	华为	Mate 7	M	19	M22-
1	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	548	Industry tag	华为	Mate 7	M	19	M22-
2	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	710	Relatives 1	华为	Mate 7	M	19	M22-
3	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	704	Property Industry 2.0	华为	Mate 7	M	19	M22-
4	6	5927333115845830913	1	1	1476664663289716375	2016-05-01 00:27:21	0.0	0.0	172	IM	华为	Mate 7	M	19	M22-

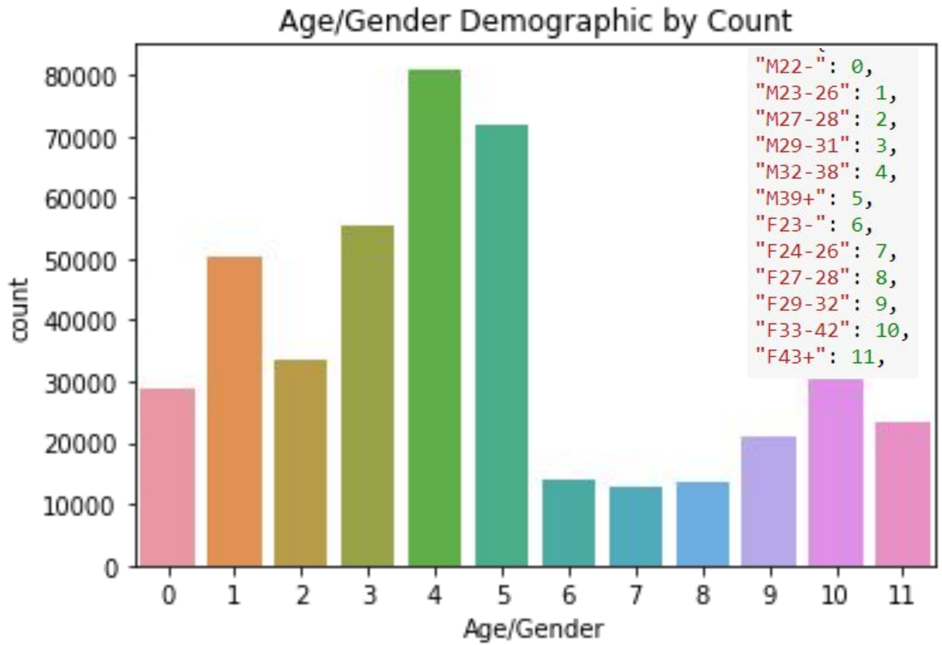
In order to condense the total rows, the label_id column was turned into a list data type

	event_id	app_id	is_active	device_id	device_model	gender	age	group	english_phone_brand	size	label_id_y
0	6	-8764672938472212518	1	1476664663289716375	Mate 7	M	19	M22-	huawei	1	[179, 548, 704, 714, 179, 548, 704, 714, 179, ...
4	6	-8271866350659046570	0	1476664663289716375	Mate 7	M	19	M22-	huawei	1	[405, 730, 737, 738, 774, 775, 780, 781, 785, ...
15	6	-7509752927626140732	0	1476664663289716375	Mate 7	M	19	M22-	huawei	1	[405, 548, 730, 756, 761, 777, 782, 787, 959, ...
26	6	-7377004479023402858	1	1476664663289716375	Mate 7	M	19	M22-	huawei	1	[183, 302, 303, 548, 549, 704, 721, 183, 302, ...
33	6	-5839858269967688123	0	1476664663289716375	Mate 7	M	19	M22-	huawei	1	[251, 254, 405, 548, 562, 564, 691, 704, 713, ...

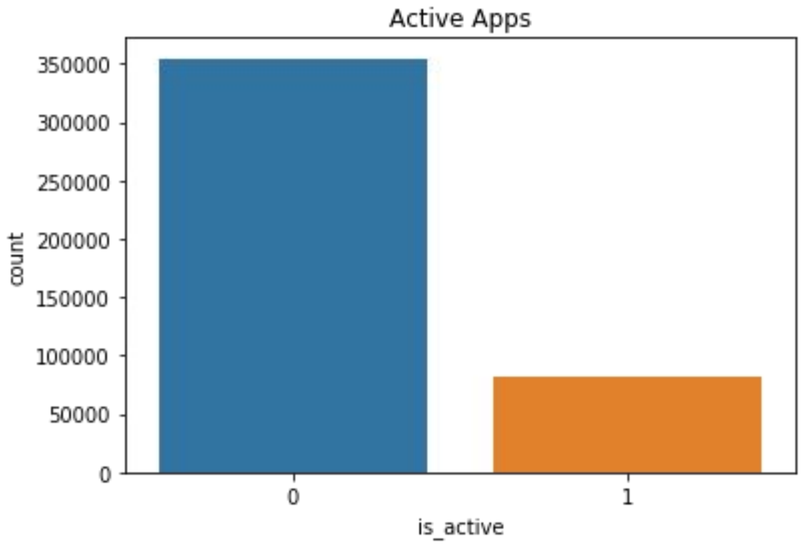
```
final_df.shape
```

```
(435990, 11)
```

- **Much more information on the M29-31 and M39+ age group**
- **Lacking Information on F23-, F24-26, and F27-28 Age group.**



- **Noticed majority of the apps are inactive, only a small portion of apps are currently active**



Why is this Important?

What can be Gained with this Information?

- Beneficial towards cell phone industry
  - Can determine if certain age groups tend to buy specific phone brands or device models.

- Beneficial towards App Industry
  - Useful to see if there are patterns between certain age groups and different app categories that each group downloads.

Classification Model Analysis

Two Classification Models Implemented

K Nearest Neighbor Classifier

- Parameters Tuned: n_neighbors, weights, and leaf size

Random Forest Classifier

- Parameters Tuned: max_depth, n_estimators, and min_samples_leaf
- Obtained most accurate score from Random Forest Classifier
    - This model was used for analysis

Dimensionality Reduction Method

- Each model was executed first without PCA
- PCA was applied after getting results of each model
- Used components that summed up to 90% variance
  - Equivalent to 403 components out of 635
- After applying PCA to each classification model, the accuracy decreased
  - Therefore, PCA was not implemented in the final classification model

Final Random Forest Classifier

- Utilized grid search and cross fold variation

```
grid = {
    "rf__max_depth": [50, 70, 90, 110],
    "rf__n_estimators": [1, 10, 100],
    "rf__min_samples_leaf": [1, 3, 5, 7],
    "rf__criterion": ["gini"],
}
```

```
{'rf__criterion': 'gini',
 'rf__max_depth': 110,
 'rf__min_samples_leaf': 1,
 'rf__n_estimators': 100}
```

Random Forest Results

	Predicted M22-	Predicted M23-26	Predicted M27-28	Predicted M29-31	Predicted M32-38	Predicted M39+	Predicted F23-	Predicted F24-26	Predicted F27-28	Predicted F29-32	Predicted F33-42	Predicted F43+
Actually M22-	3118	453	212	383	600	400	122	59	83	102	120	89
Actually M23-26	420	5499	377	732	1187	851	140	132	126	181	253	182
Actually M27-28	238	477	3150	612	848	670	62	87	86	149	213	143
Actually M29-31	281	668	441	6284	1288	1065	85	124	118	175	334	221
Actually M32-38	380	919	534	943	10278	1573	138	117	177	291	495	346
Actually M39+	264	639	385	878	1596	9271	114	112	135	220	485	293
Actually F23-	154	231	83	148	278	186	1357	71	31	73	98	60
Actually F24-26	119	174	114	189	268	264	83	1049	62	80	101	59
Actually F27-28	120	207	95	215	385	268	35	49	1079	70	127	82
Actually F29-32	151	314	186	279	530	485	60	64	64	1746	163	128
Actually F33-42	180	297	199	424	807	806	83	80	89	144	2770	180
Actually F43+	104	302	217	336	631	602	59	61	62	138	203	1967

	precision	recall	f1-score	support
0	0.56	0.54	0.55	5741
1	0.54	0.55	0.54	10080
2	0.53	0.47	0.49	6735
3	0.55	0.57	0.56	11084
4	0.55	0.63	0.59	16191
5	0.56	0.64	0.60	14392
6	0.58	0.49	0.53	2770
7	0.52	0.41	0.46	2562
8	0.51	0.39	0.45	2732
9	0.52	0.42	0.46	4170
10	0.52	0.46	0.49	6059
11	0.52	0.42	0.47	4682
accuracy			0.55	87198
macro avg	0.54	0.50	0.52	87198
weighted avg	0.54	0.55	0.54	87198

Train score: 0.9999971329617652

Test score: 0.5455170990160325

Clustering and Dimensionality Reduction Methods

Clustering Algorithms: KMeans vs. Gaussian Mixture Model

KMeans

- Tuned n_cluster parameter to include a range of 3 - 10 clusters
- Applied PCA and UMAP to clustering

Gaussian Mixture Model

- Tuned n_components parameter to include range of 3 - 10 clusters
- Applied PCA and UMAP to clustering

PCA vs. UMAP

PCA

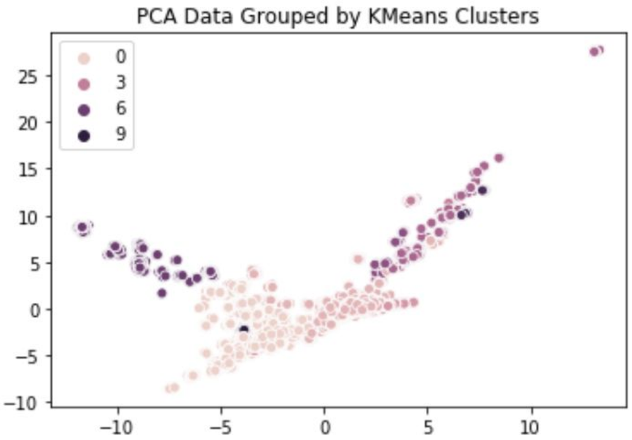
- Compared Silhouette Scores for PCA n_components equating to 60%, 70%, 80%, and 90% variance
- Best results were obtained when amount of components were equivalent to 60% variance
    - Resulted in using 211 components

UMAP

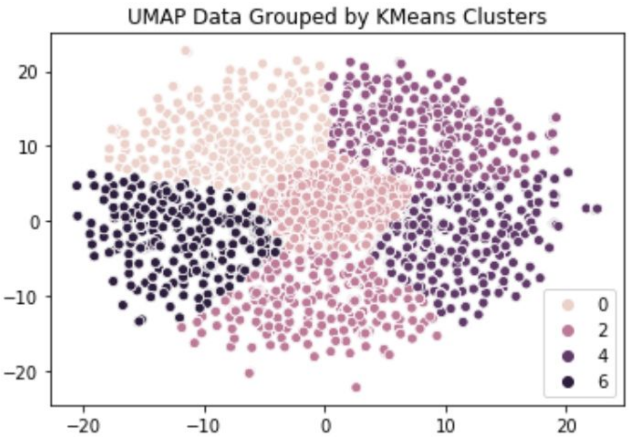
- Compared Silhouette Scores for UMAP n_neighbors equating to 5, 10, 100, and 200
- Best results came from n_neighbors = 200

PCA vs. UMAP

- Highest Silhouette Score of 0.098 with KMeans and 9 clusters

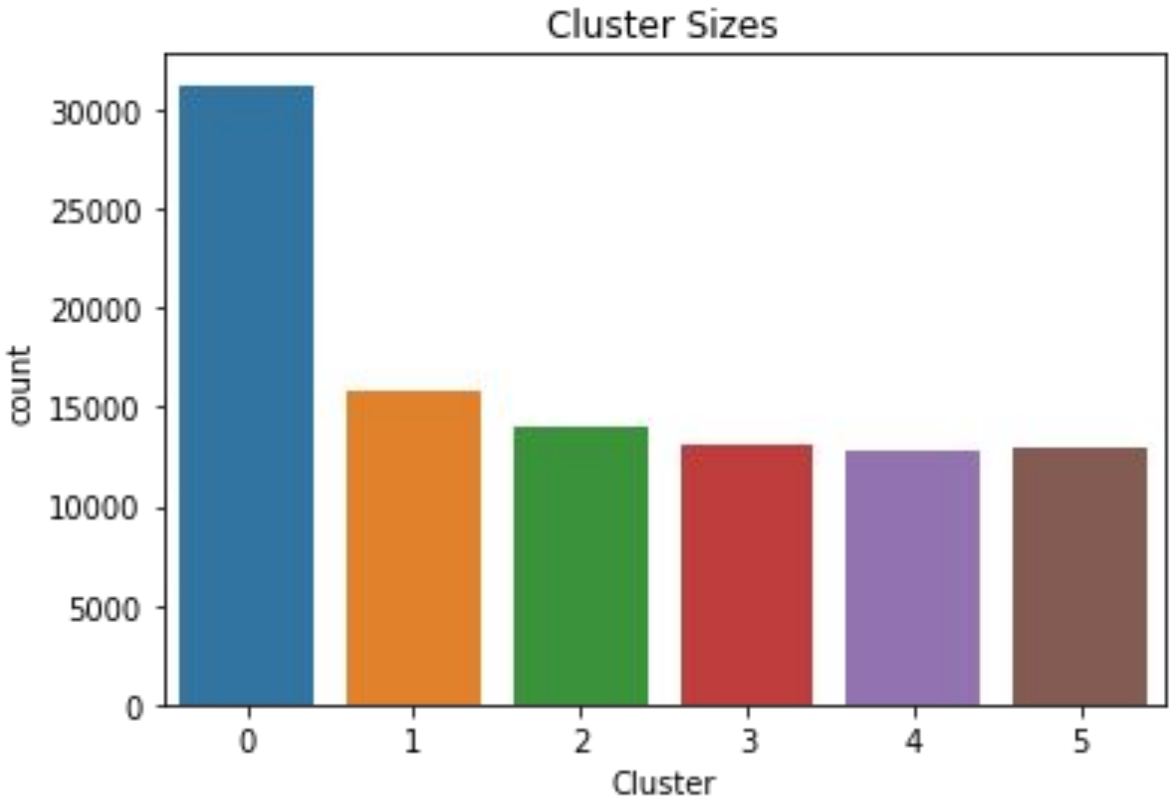


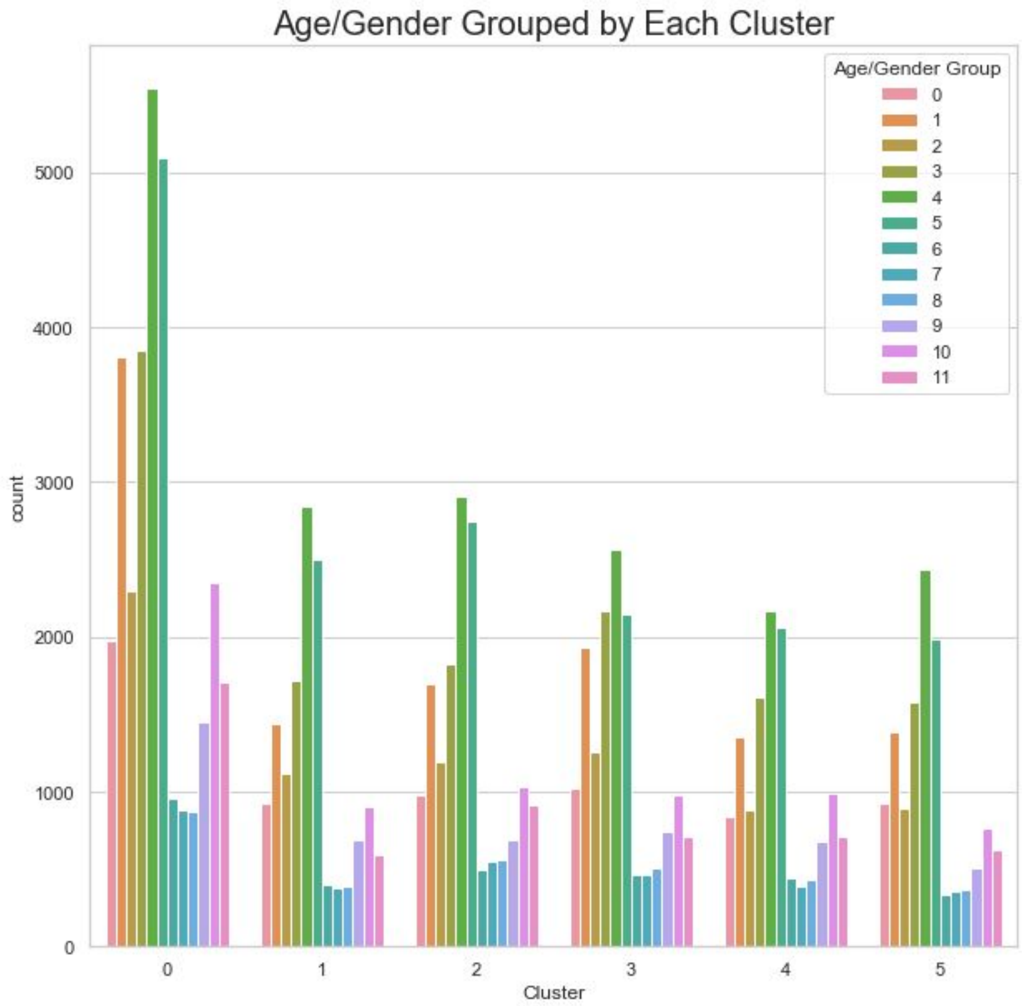
- Highest Silhouette Score of 0.360 with KMeans and 6 clusters



KMeans with UMAP Results

0	31259
1	15821
2	14069
3	13104
5	13007
4	12740




Age/Gender Grouped by Each Cluster
Age/Gender Group
0
1
2
3
4
5
6
7
8
9
10
11
count
0
1000
2000
3000
4000
5000
Cluster
0
1
2
3
4
5

Neural Network Classification

Neural Network Parameter Tuning

- Activation Functions Used:
  - RELU
  - Tanh
  - Sigmoid
- Consistently used 3 layers
- Optimizers Used:
  - Adam
  - SGD
- Increasing epochs
  - 20, 50, 100

Seven Models Created

- 3 Layer RELU Model with ADAM Optimizer
- 3 Layer RELU Model with SGD Optimizer
- 3 Layer TANH Model with ADAM Optimizer
- 3 Layer TANH Model with SGD Optimizer
- 3 Layer Sigmoid with ADAM Optimizer
- 3 Layer Sigmoid with SGD Optimizer
- 3 Layer RELU with ADAM Optimizer, Smaller Batch Size, and More Epochs

Neural Network Results

Model Results

- Used accuracy metric to compare each model
  - Accuracy of each model was relatively low (between 0.15-0.19)
- Best model was obtained from 3 Layer RELU Model with ADAM Optimizer
  - Accuracy of 0.19
  - Increased epochs
  - Decreased batch size
  - Changing these parameters did not affect the accuracy

Model Results

Predicted

	0	0	0	0	5587	154	0	0	0	0	0	0
	0	0	0	0	9811	269	0	0	0	0	0	0
	0	0	0	0	6559	176	0	0	0	0	0	0
	0	0	0	0	10836	248	0	0	0	0	0	0
	0	0	0	0	15790	401	0	0	0	0	0	0
Actual	0	0	0	0	14007	385	0	0	0	0	0	0
	0	0	0	0	2705	65	0	0	0	0	0	0
	0	0	0	0	2501	61	0	0	0	0	0	0
	0	0	0	0	2665	67	0	0	0	0	0	0
	0	0	0	0	4077	93	0	0	0	0	0	0
	0	0	0	0	5927	132	0	0	0	0	0	0
	0	0	0	0	4552	130	0	0	0	0	0	0

Limitations and Areas to be Improved

Supervised/Unsupervised Learning Limitations

- Only used two classification models
- Only used two clustering models
- Only used PCA for supervised learning
- Only used PCA and UMAP for unsupervised learning
- Had to implement subsample of data and apply models to subsamples
- These constraints appear to have impacted the performance of the models

Neural Network Limitations

- Only implemented 3 layer models
- Kept learning rate and momentum for loss functions constant

How to Improve Models

- Fit entire data set for supervised and unsupervised learning models
- Incorporate SelectKBest for supervised learning models
- Incorporate t-SNE for unsupervised learning models
- Create a Convolutional Neural Network
- Tune momentum and learning rate in loss functions

Thank You!

Questions or Comments?