Textual analysis of movie reviews Team5

BY

HARSH PATHAK

JIDAPA THADAJARASSIRI

KRUSHIKA TAPEDIA

PITCHAYA WIRATCHOTISATIAN

PRINCE SHIVA CHAUDHARY

Emotions

http://webneel.com/daily/20-inside-out-characters
http://www.thecoli.com/threads/ios-emojis-degrade-and-simplify-human-expression-of-emotion.51174/
https://www.willbrattcounselling.com/blog-creating-difference/2015/1/12/your-emotions-arent-a-problem

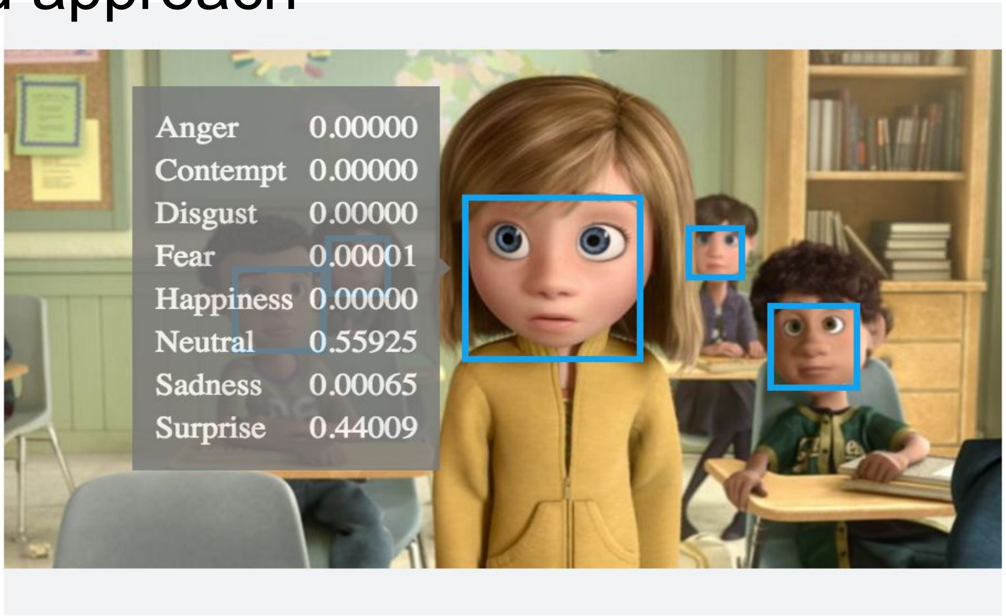
Emotions are difficult to express in words

Image Courtesy: https://twitter.com/AnupKaphle

Machine Learning to understand Emotions

Two Approaches

- Lexicon based approach
- Machine learning based approach

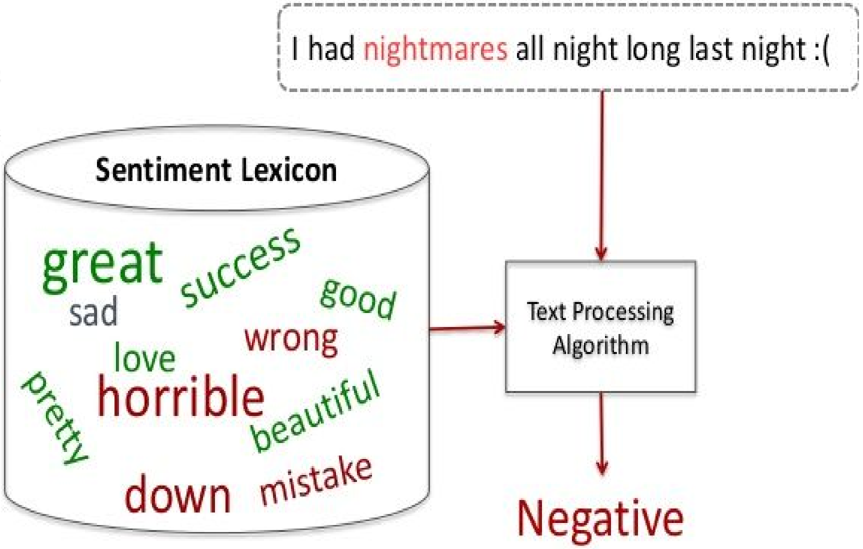


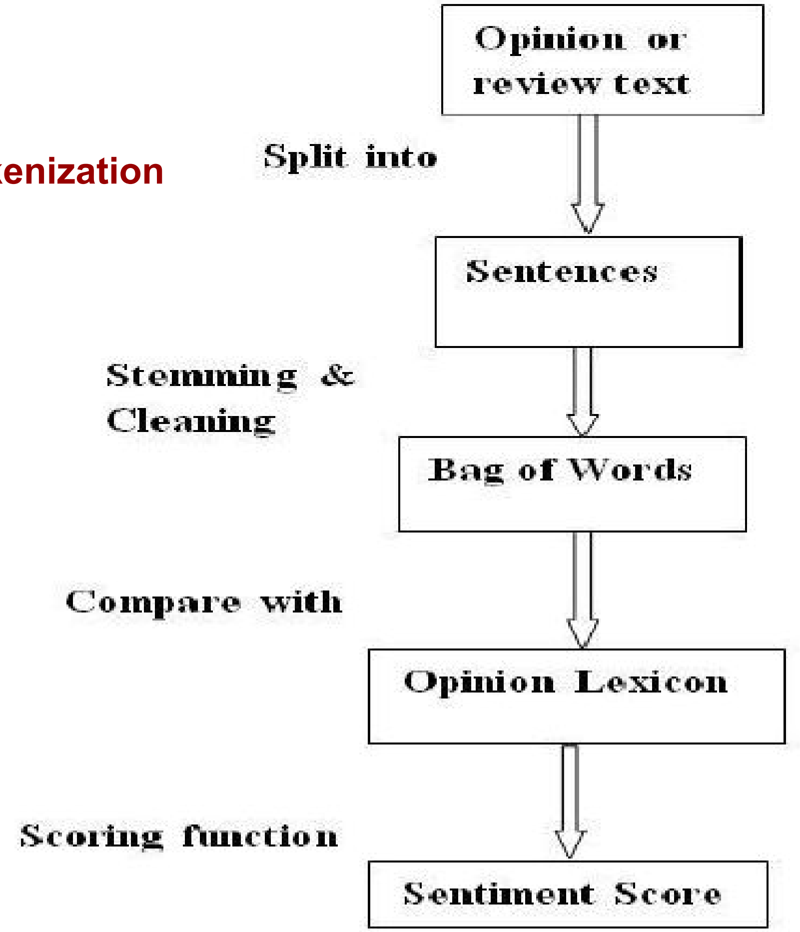
http://thenextweb.com

Approach

Lexicon-based

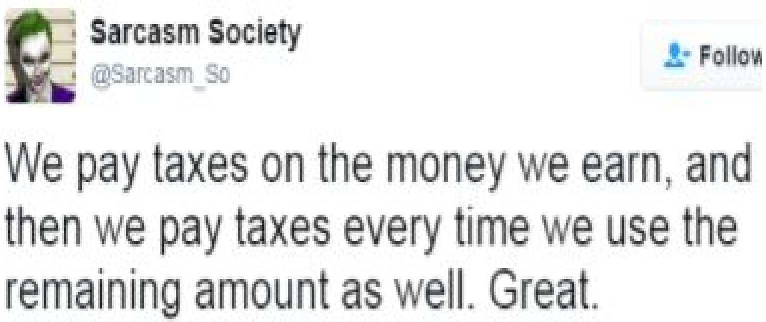
Tokenization



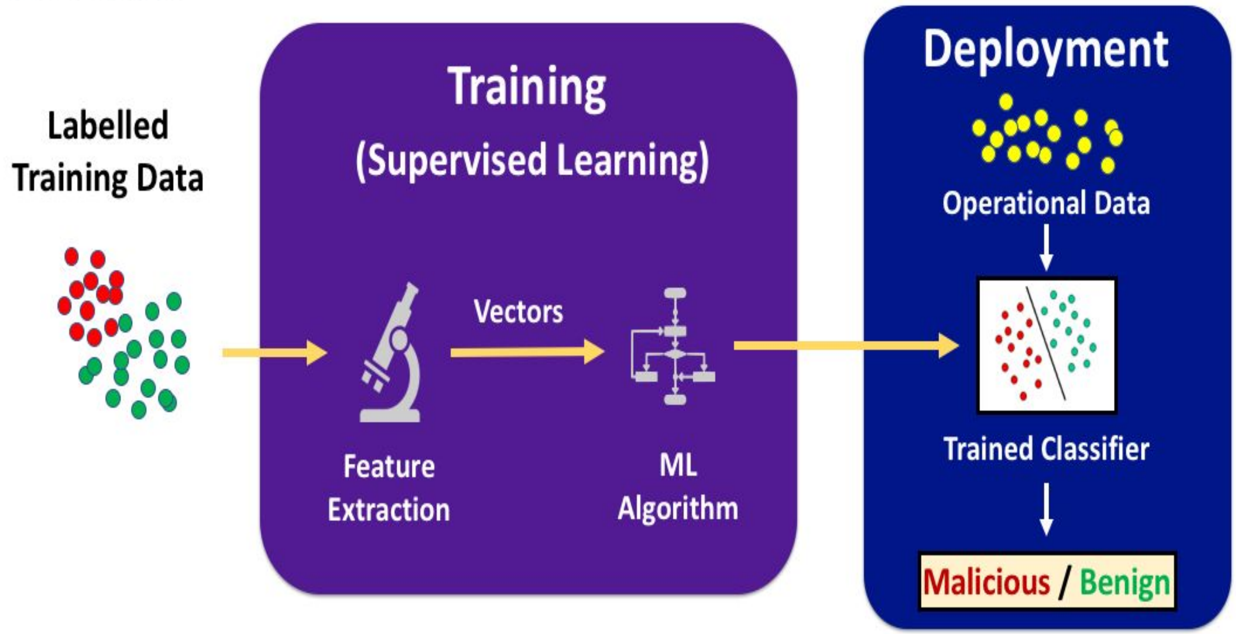


https://www.slideshare.net/Staano/senticircles-for-contextual-and-conceptual-semantic-sentiment-analysis-of-twitter

Machine Learning Approach


Sarcasm Society
@Sarcasm_So
Follow

We pay taxes on the money we earn, and then we pay taxes every time we use the remaining amount as well. Great.



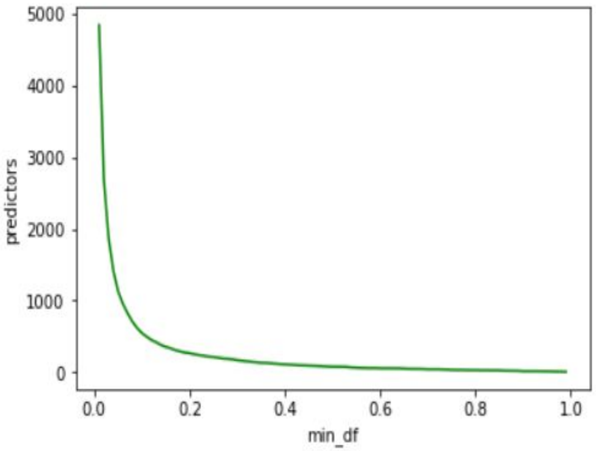
https://www.researchgate.net/figure/221561415_fig1_Figure-1-Supervised-Machine-Learning-Schema

Introduction

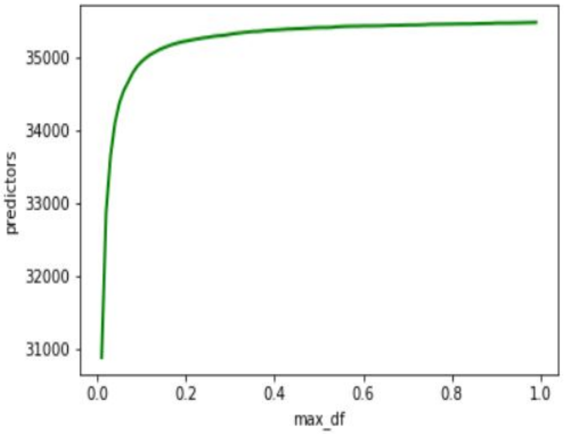
I. Sentiment Analysis

II. TfidfVectorizer

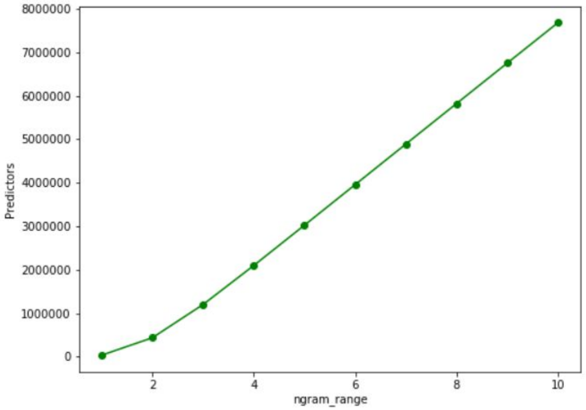
min_df



max_df

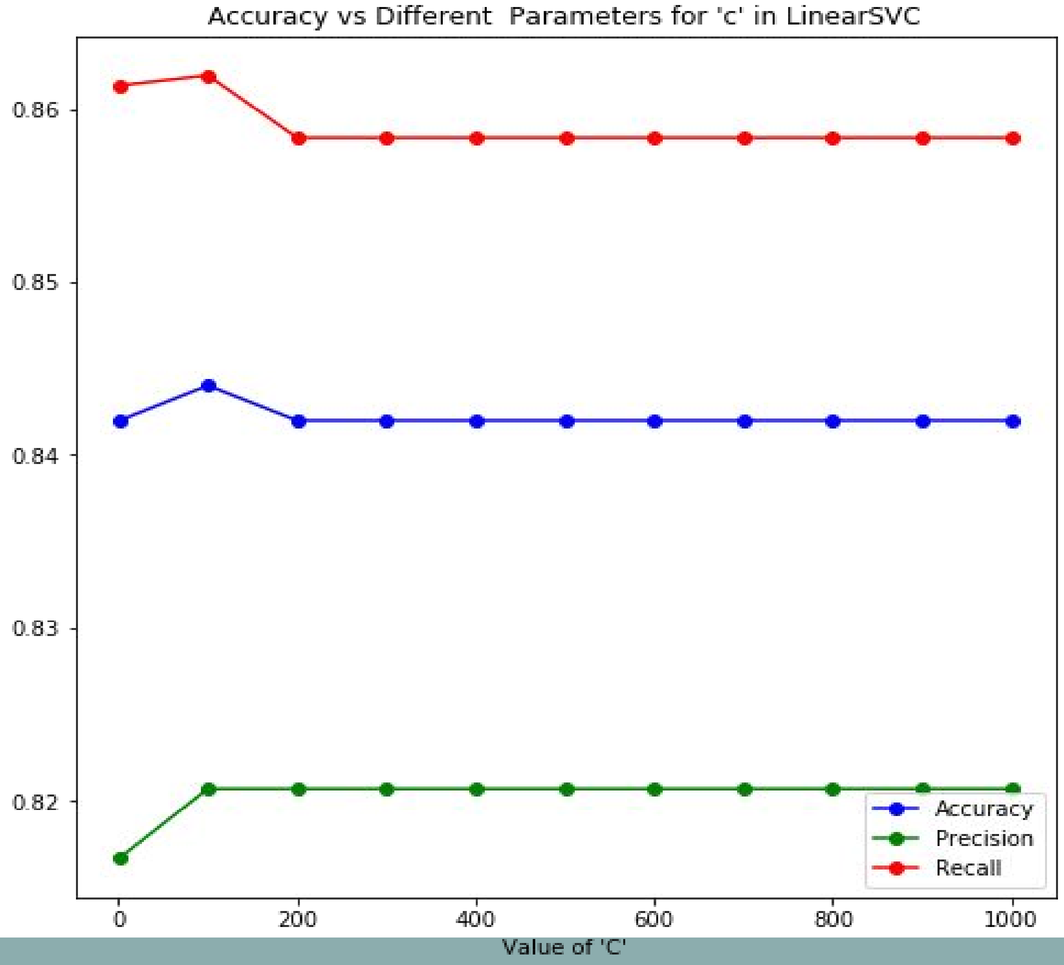


ngram_range



Machine Learning Algorithms (LinearSVC)

Performance of LinearSVC : Parameter C

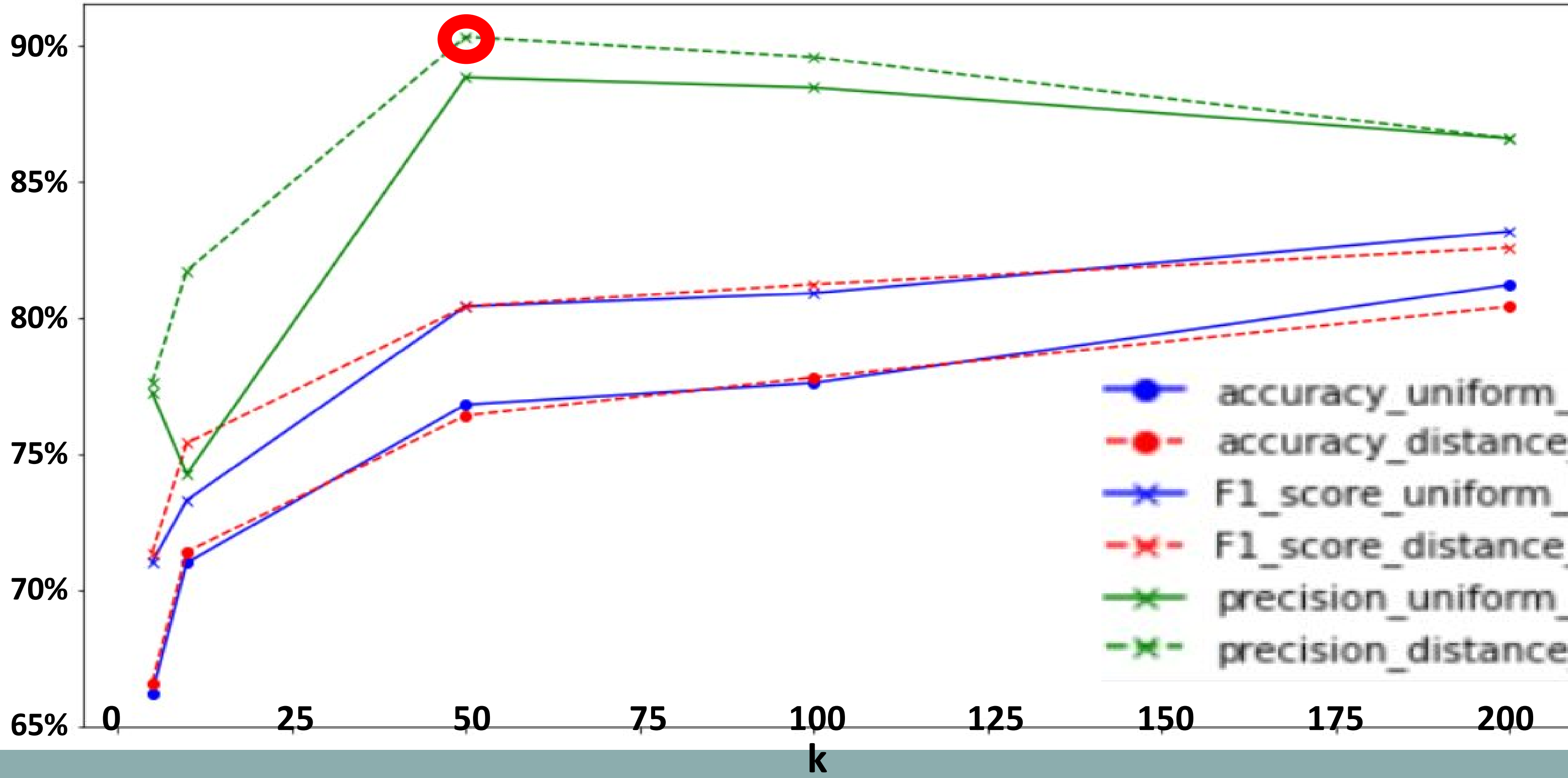


Machine Learning Algorithms (kNN)

Performance of KneighborClassifier

Parameter : k = 5, 10, 50, 100, 200

Parameter : weight function = ‘uniform’ or ‘distance’



Finding the right plot (1)

Step 1: Data Preprocessing

- Remove stopwords : NLTK
- Extract stem-words : snowball
- Create TF-IDF vector matrix : TfidfVectorizer

Step 2: Feature Selection

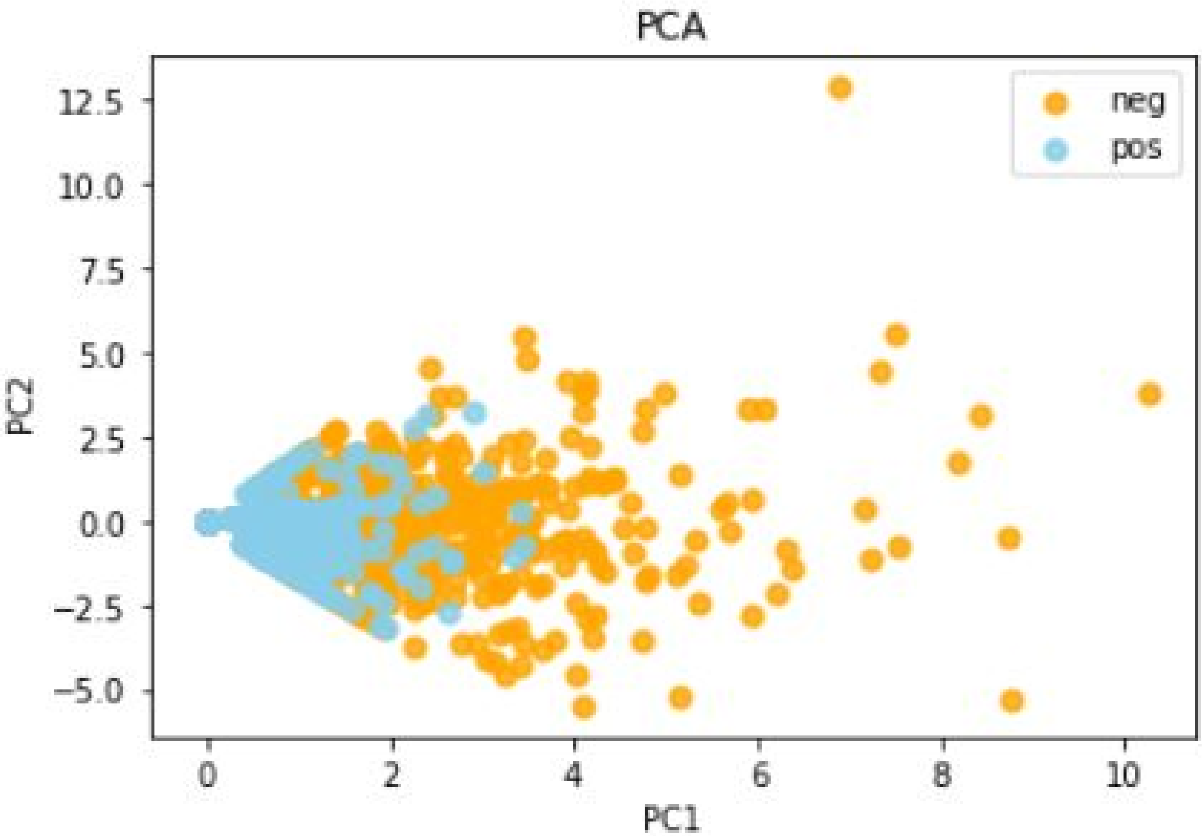
- Logistic regression with lasso
- Linear model with lasso
- TruncatedSVD: first 2 components, 256 components

Step 3: Methodologies

1. Feature selection: TruncatedSVD, Lasso
2. Clustering: K-Means clustering, Hierarchical clustering
3. Ensemble learning: RandomTreesEmbedding
4. Manifold learning: MDS, Isomap, Spectral decomposition, Locally Linear Embedding, t-SNE

Finding the right plot (2)

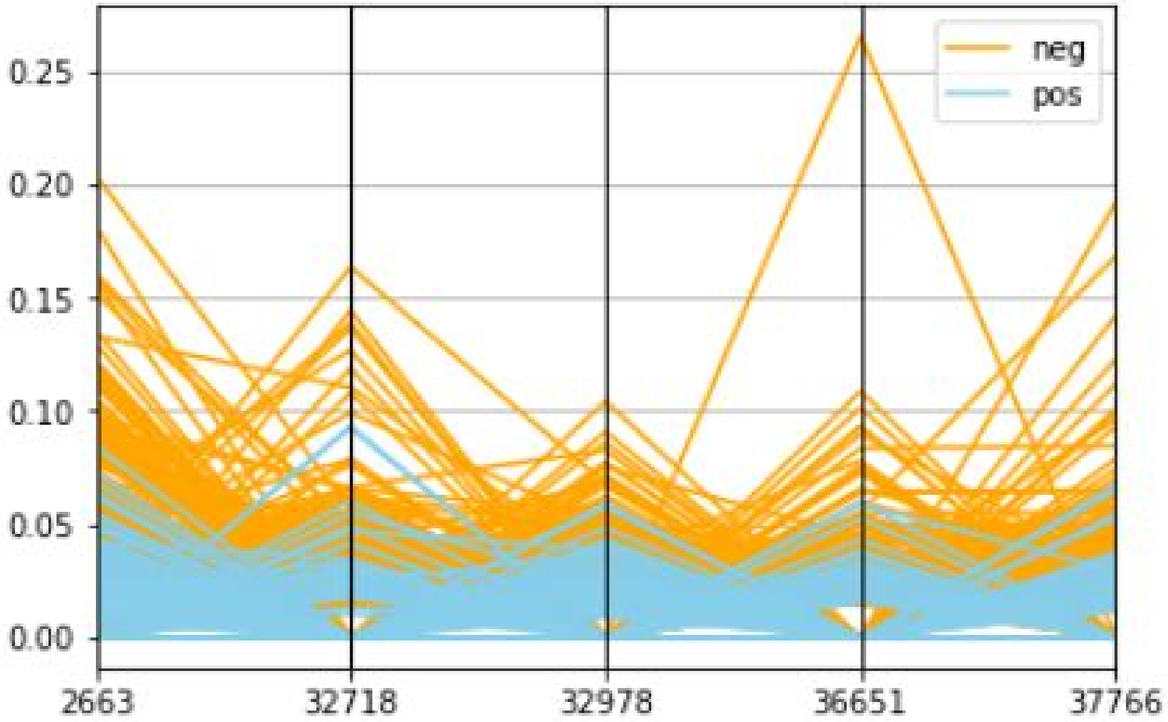
Method 1: TruncatedSVD



The first two PCs of features in LinearLassoIndex

Finding the right plot (3)

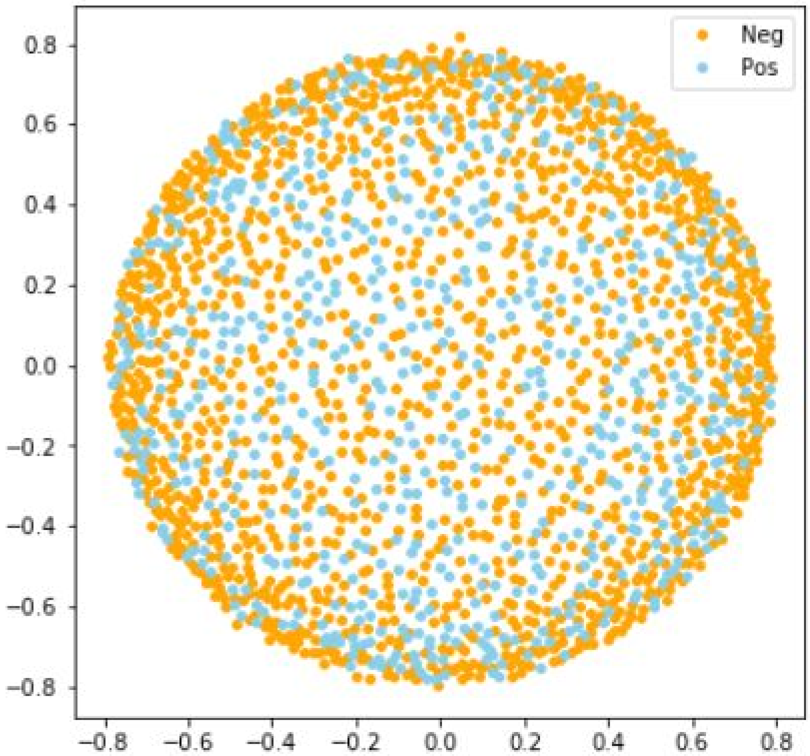
Method 2: Lasso



Parallel coordinates plot of features from LinearLassoIndex

Finding the right plot (4)

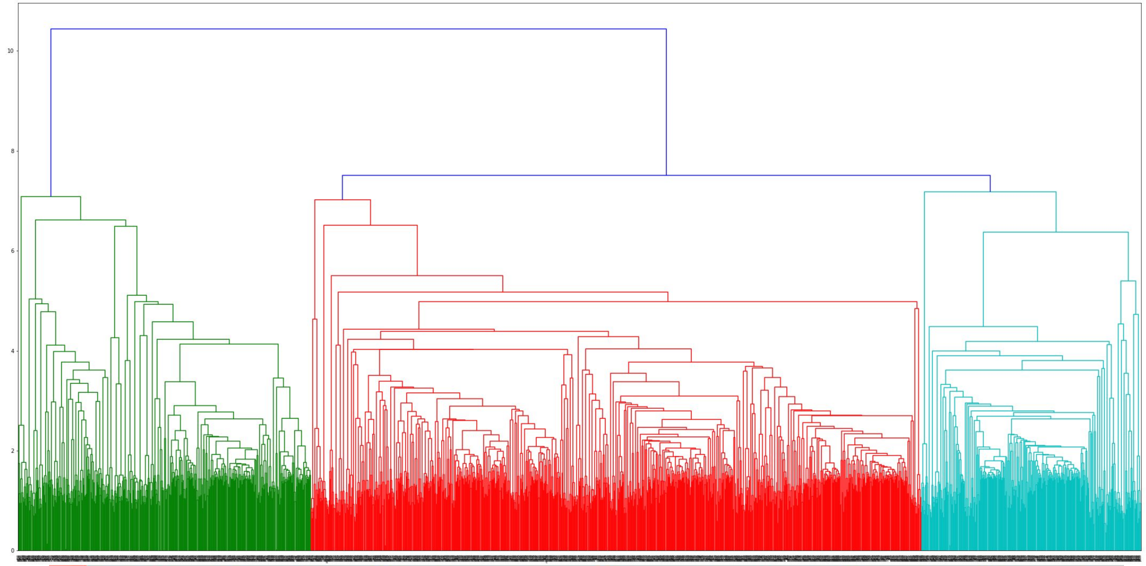
Method 3: K-Means Clustering



K-mean of the MDS(n_components =2) of the distance matrix where n_clusters = 2

Finding the right plot (5)

Method 4: Hierarchical Clustering



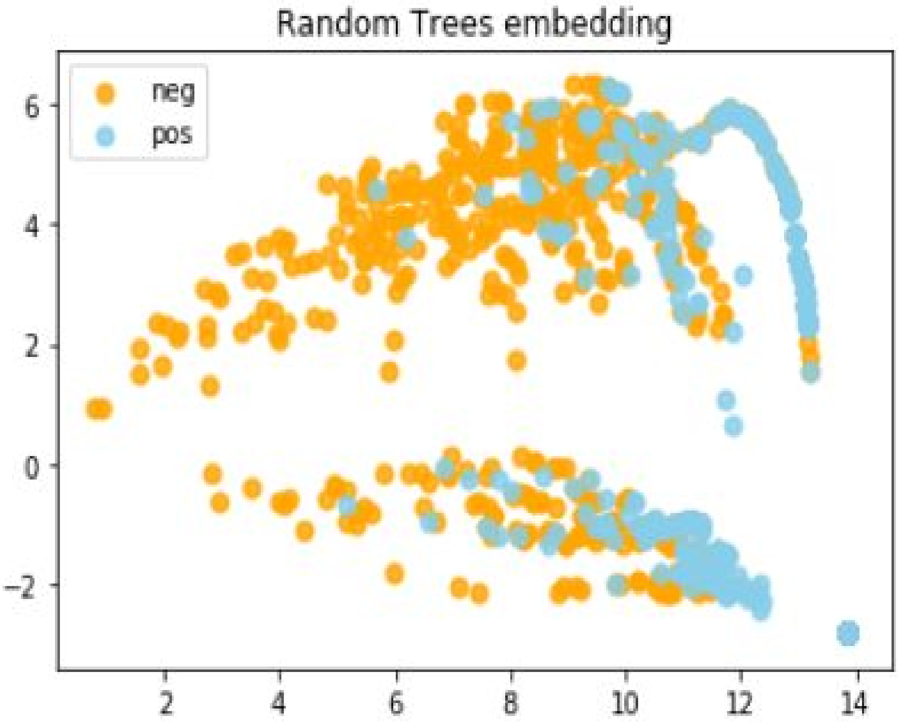

```
['p434', 'p819', 'p823', 'p232', 'p329', 'p415', 'p613', 'p75', 'p10', 'p398', 'p505', 'p199', 'p785',
 'p707', 'p958', 'p425', 'n67', 'p136', 'p403', 'n726', 'p5', 'n548', 'p403', 'p505', 'p370', 'p696',
'p179', 'p1000', 'p265', 'p322', 'p188', 'p837', 'p749', 'p79', 'p283', 'p965', 'n365', 'n935', 'n390',
'p534', 'n370', 'n582', 'n151', 'n23', 'n134', 'n347', 'n947', 'n220', 'n349', 'n887', 'n716', 'p197',
'p473', 'n313', 'n847', 'p31', 'p600', 'p476', 'n188', 'n322', 'n265', 'p579', 'n494', 'n449', 'n984',
```

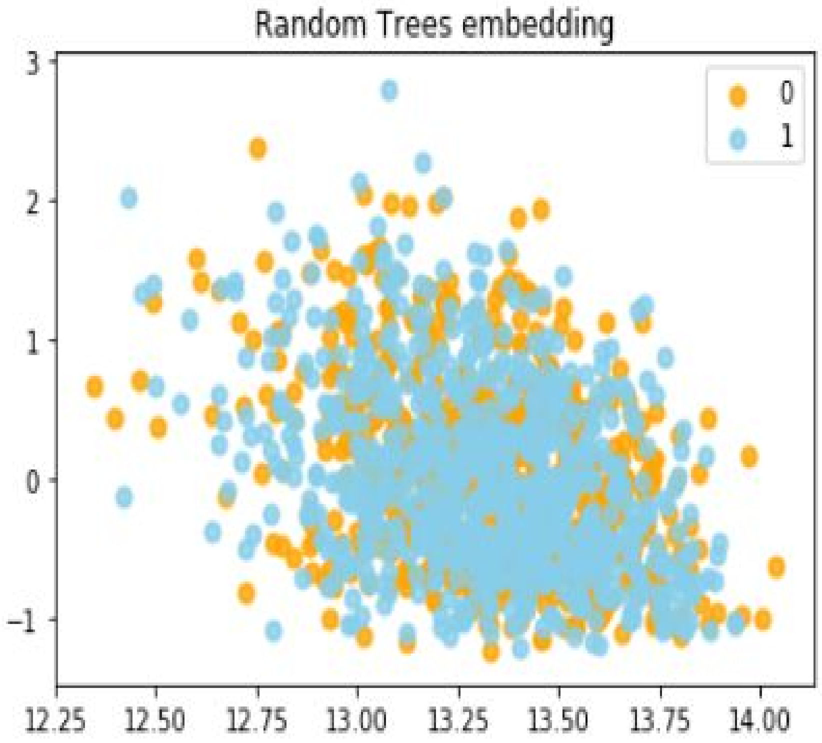
```
      ['n401', 'n999', 'n945', 'p612', 'n760', 'p647', 'n249', 'n457', 'p832', 'p496', 'p78', 'n764',
 'p433', 'n52', 'p341', 'n243', 'n731', 'n819', 'p168', 'p752', 'p66', 'p130', 'p326', 'p543', 'n195',
 'p17', 'n302', 'p589', 'p727', 'n590', 'p140', 'p365', 'p922', 'n13', 'n921', 'n294', 'n566', 'n247',
'n673', 'n738', 'n143', 'p575', 'n89', 'n230', 'n407', 'n343', 'p238', 'n397', 'n204', 'p495', 'p779',
'p134', 'p858', 'p531', 'n520', 'n854', 'n914', 'p210', 'p160', 'p566', 'p13', 'n364', 'n394', 'n617',
```

```
['p412', 'p954', 'p225', 'p62', 'p825', 'p715', 'p971', 'p38', 'p458', 'p989', 'n604', 'n927', 'p947',
   'p668', 'p147', 'n87', 'p32', 'p273', 'p346', 'p86', 'n558', 'p298', 'p212', 'p487', 'p512', 'n92',
 'p607', 'p649', 'p290', 'n261', 'p22', 'n768', 'n444', 'p816', 'n541', 'p26', 'p452', 'n580', 'n773',
 'n80', 'n957', 'n527', 'n538', 'n389', 'n662', 'n295', 'n72', 'p945', 'p538', 'n136', 'n101', 'n903',
    'n642', 'n71', 'n626', 'p784', 'n177', 'p933', 'n441', 'n16', 'n6', 'n129', 'n330', 'n27', 'n368',
 'n416', 'n593', 'n438', 'n711', 'n170', 'n190', 'n18', 'p493', 'p573', 'p694', 'p50', 'p893', 'p349',
```

Finding the right plot (6)

Method 5: RandomTreesEmbedding



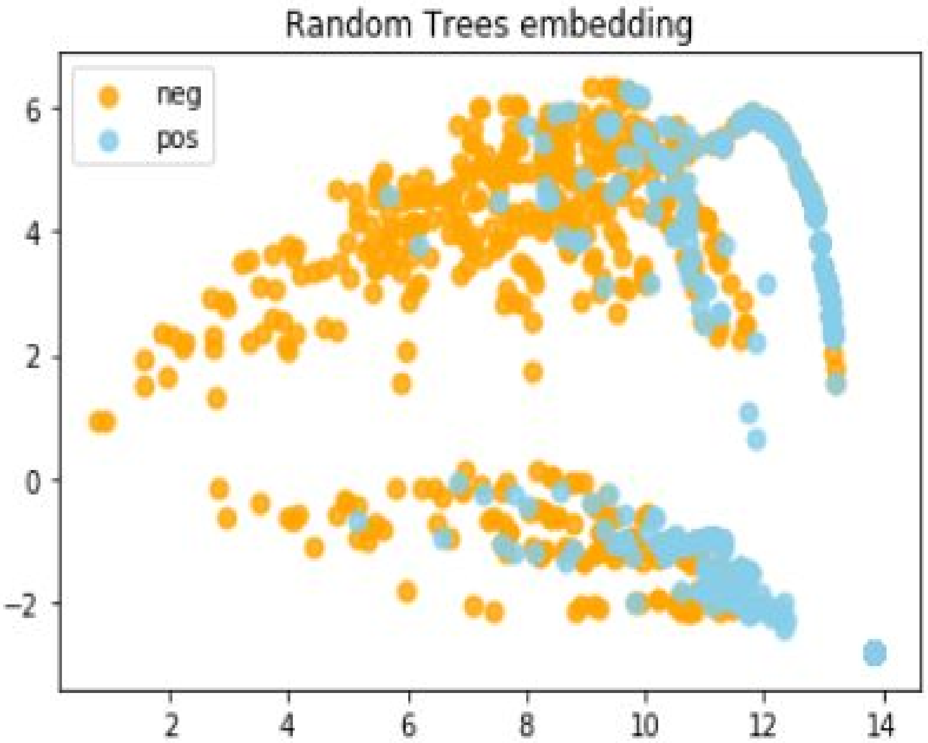


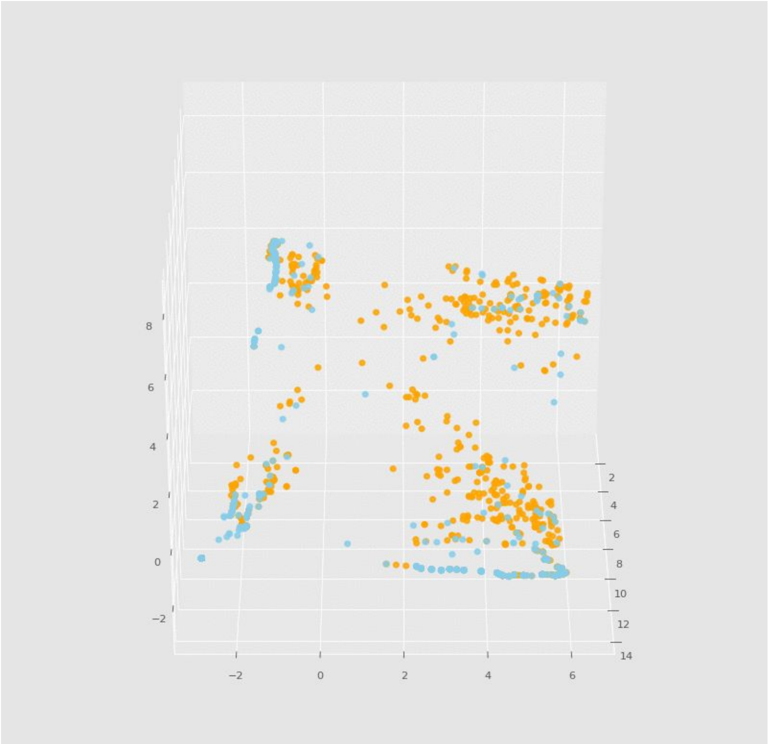
n_estimators = 200, max_depth = 5, and features in LinearLassoIndex

TruncatedSVD(256)

Finding the right plot (6)

Method 5: RandomTreesEmbedding



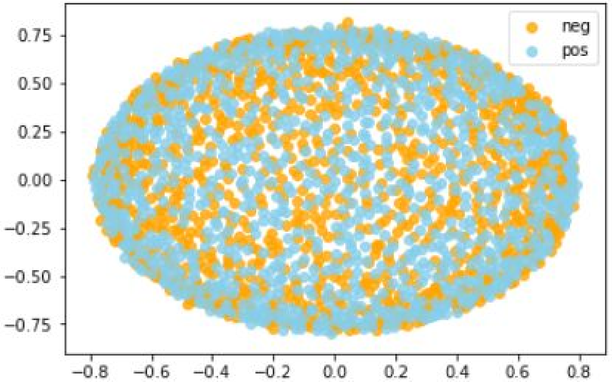


n_estimators = 200, max_depth = 5, and features
in LinearLassoIndex

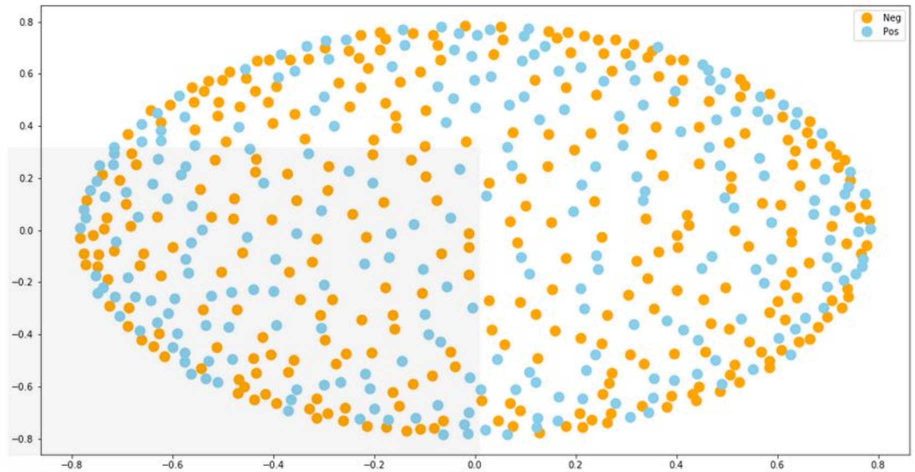
Finding the right plot (7)

Method 6: MDS

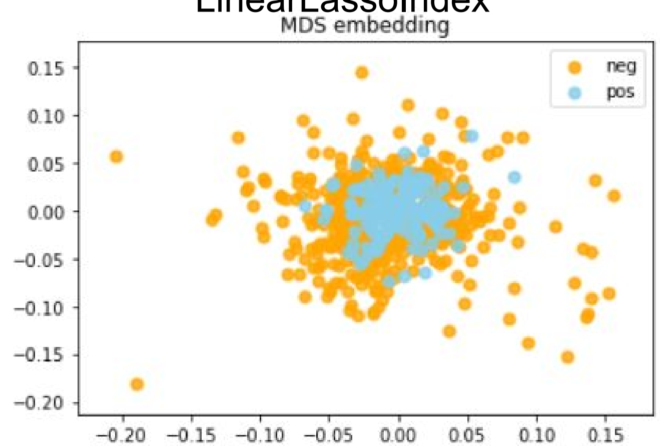
Distance matrix computed from cosine similarity



TruncatedSVD(256)

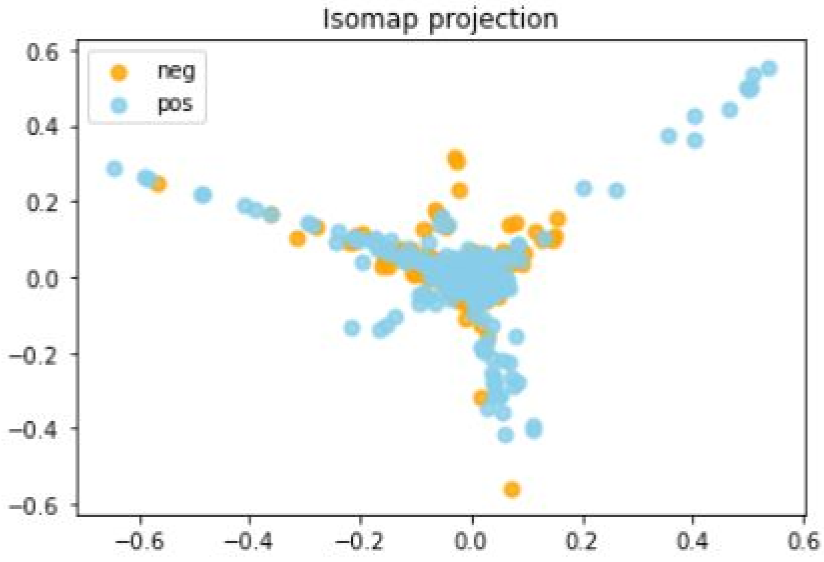


MDS applied to features in LinearLassoIndex

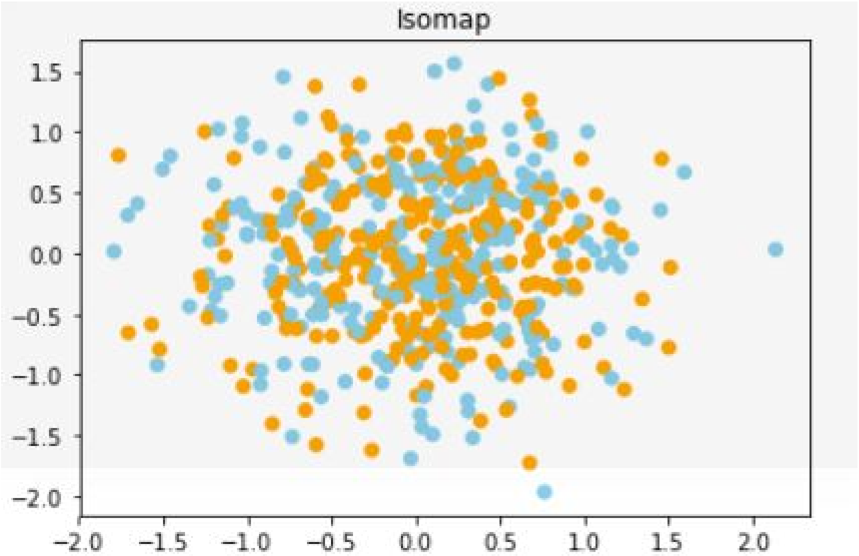


Finding the right plot (8)

Method 7: Isomap Projection



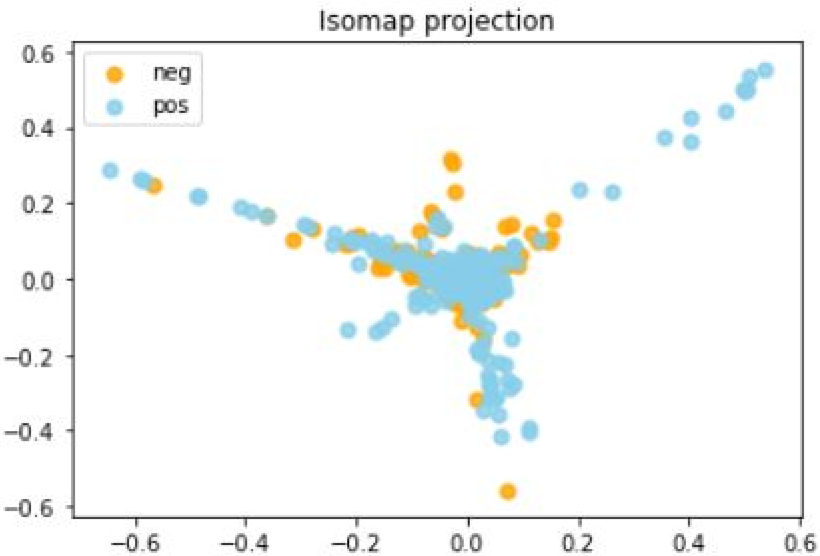
n_neighbors = 100, n_components = 2, and features in LogRegLassoIndex

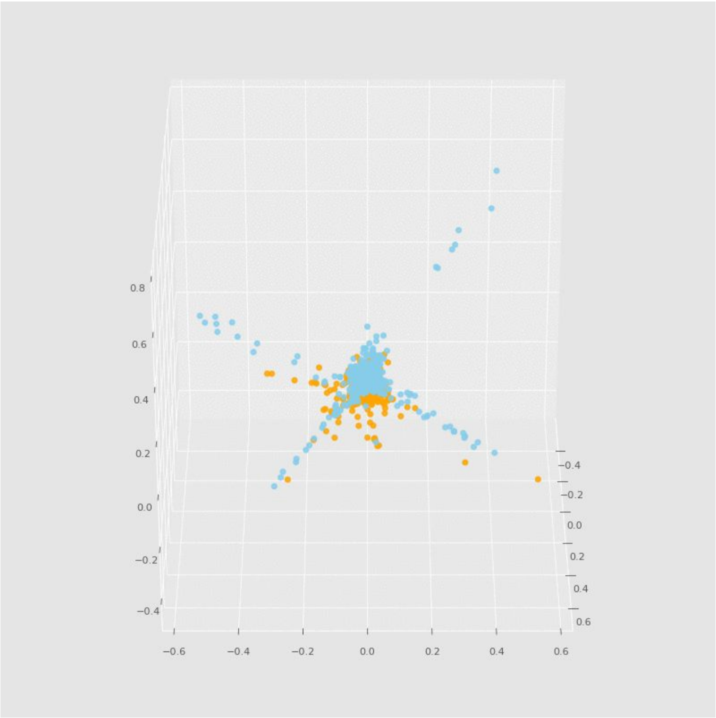


TruncatedSVD(256)

Finding the right plot (8)

Method 7: Isomap

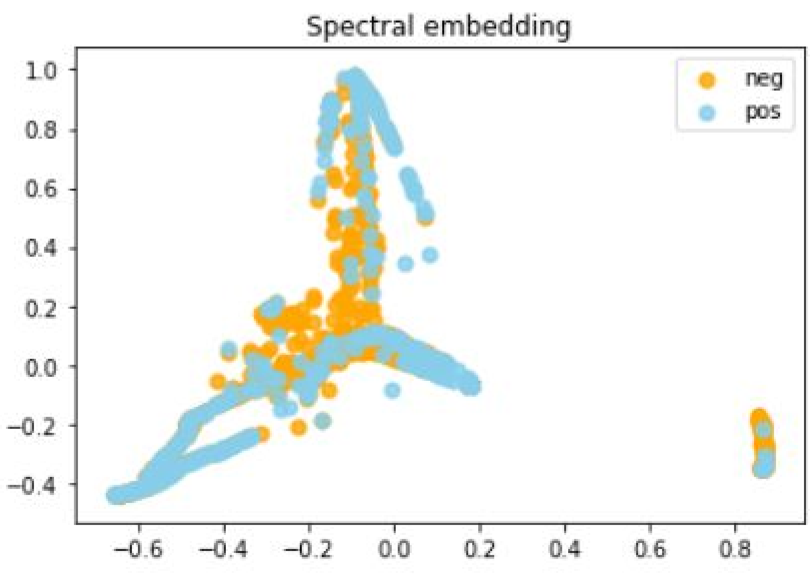




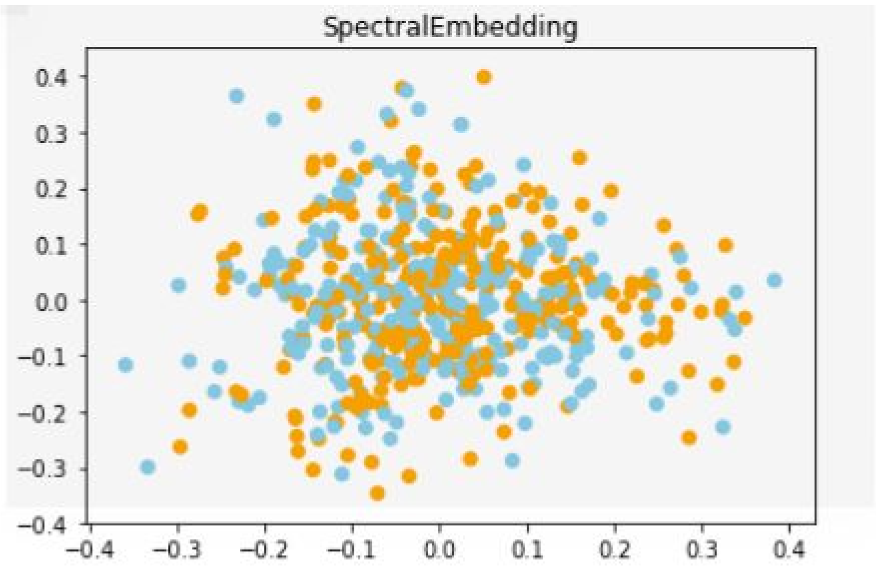
n_neighbors = 100, n_components = 2, and features in LogRegLassoIndex

Finding the right plot (9)

Method 8: Spectral Embedding



eigen_solver = ‘arpack’ and
features in LinearLassoIndex



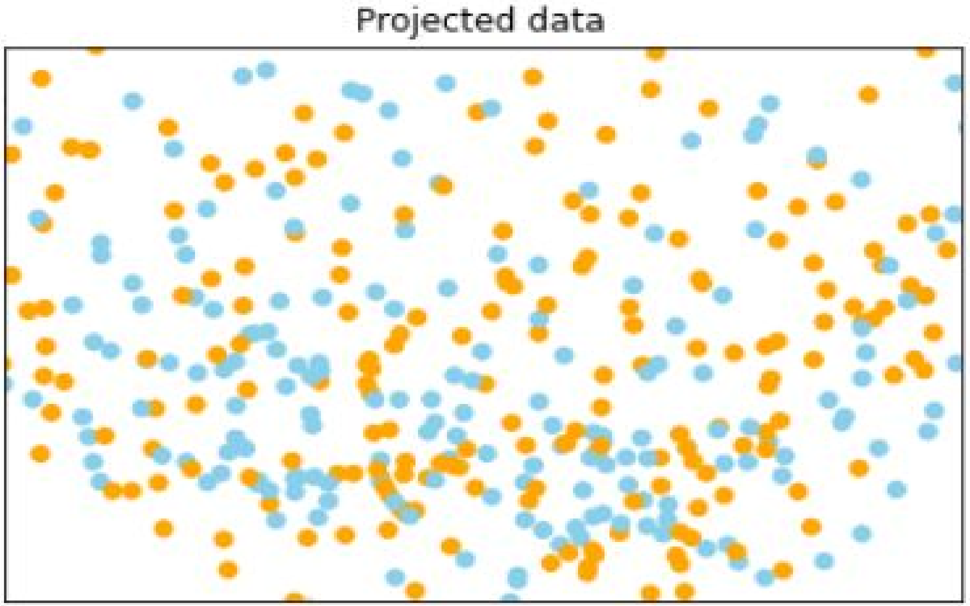
TruncatedSVD(256)

Apply t-SNE with init = ’pca’ to the data matrix

Finding the right plot (10)

Method 9: Locally Linear Embedding

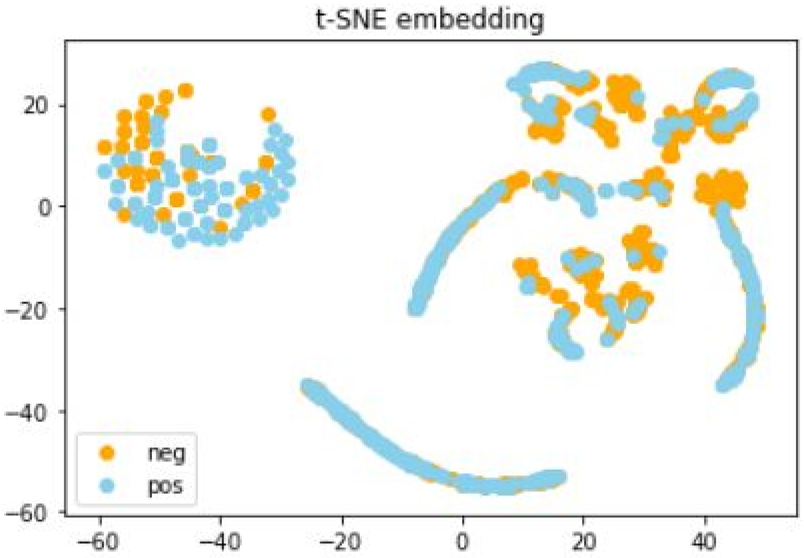
manifold.locally_linear_embedding(data_svd, n_neighbors=2, n_components=2)



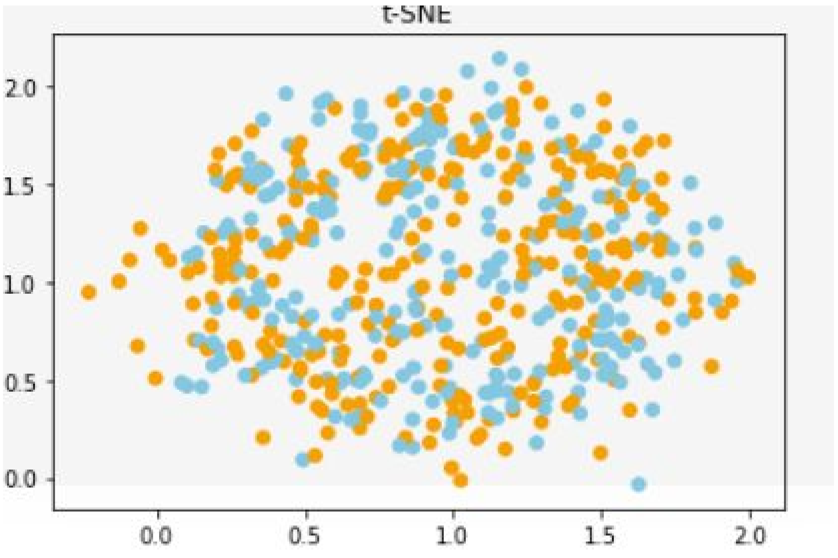
Finding the right plot (11)

Method 10: t-SNE

init = 'pca' and features in LinearLassoIndex



TruncatedSVD(256)



Thank you

Q & A