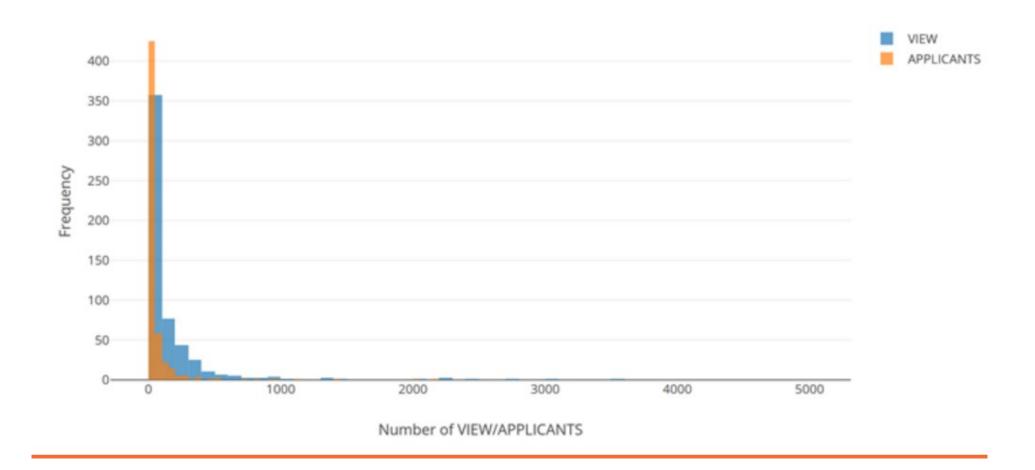
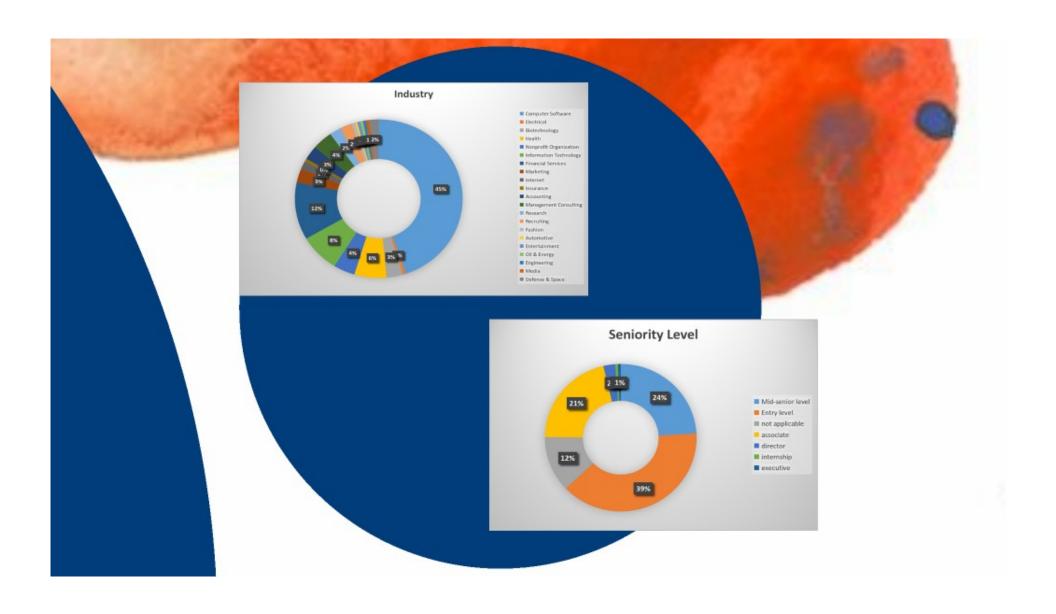
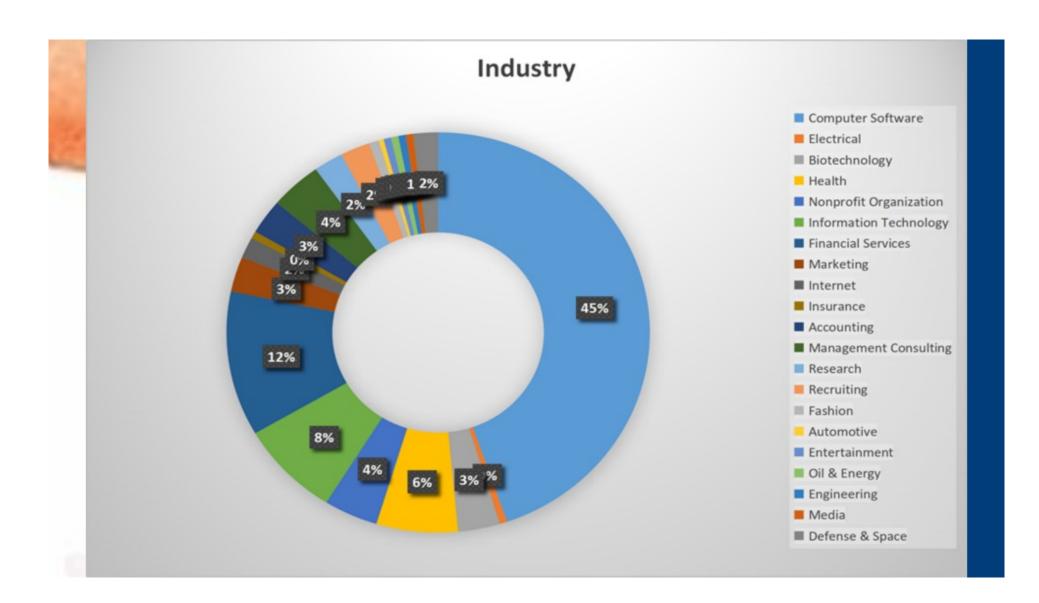
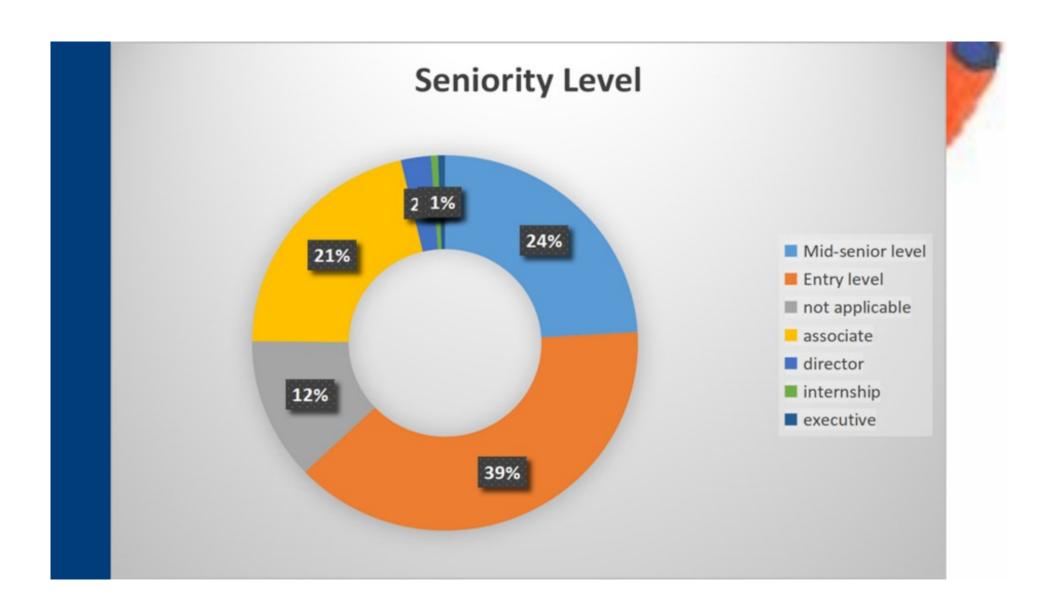


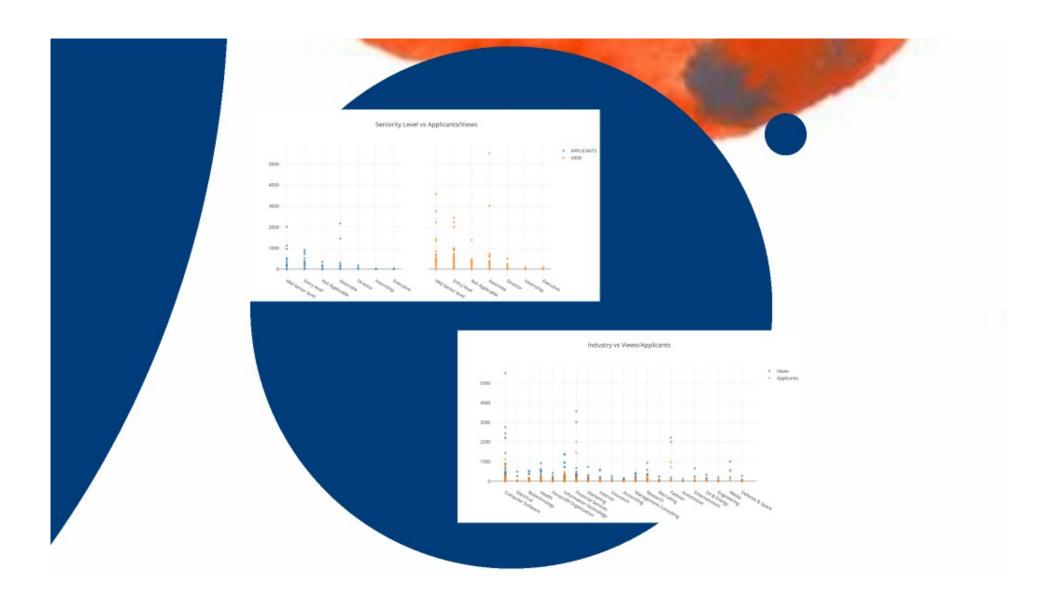
Comparison between VIEW and APPLICANTS



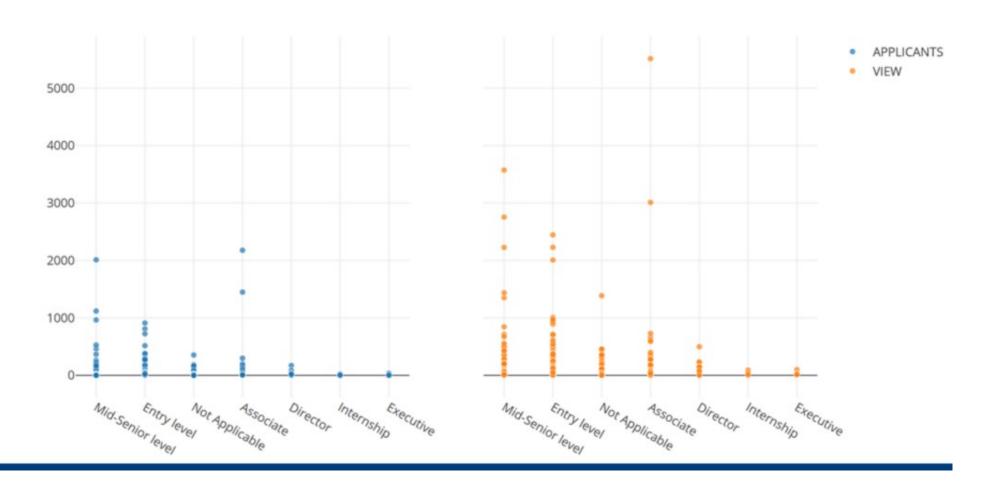




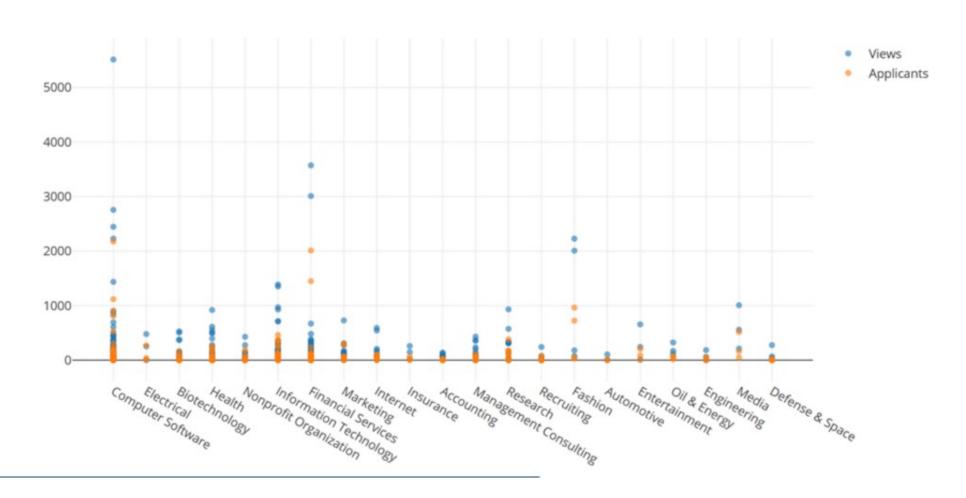


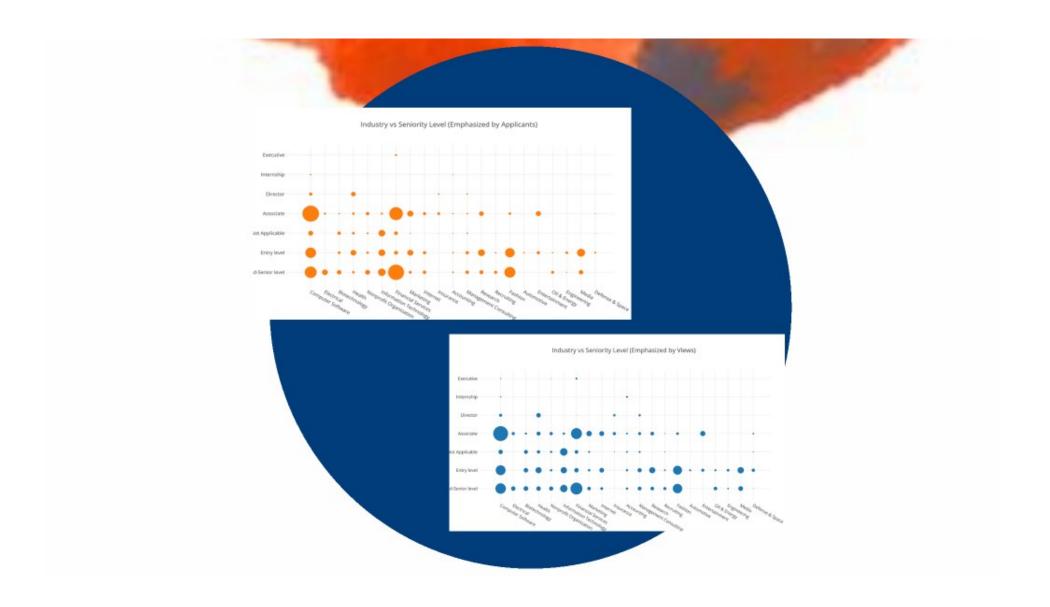


Seniority Level vs Applicants/Views

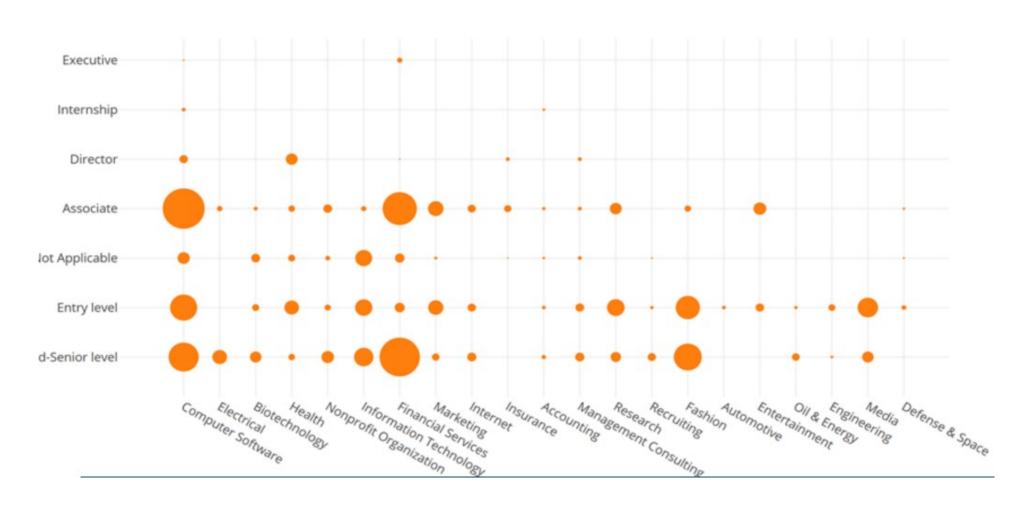


Industry vs Views/Applicants

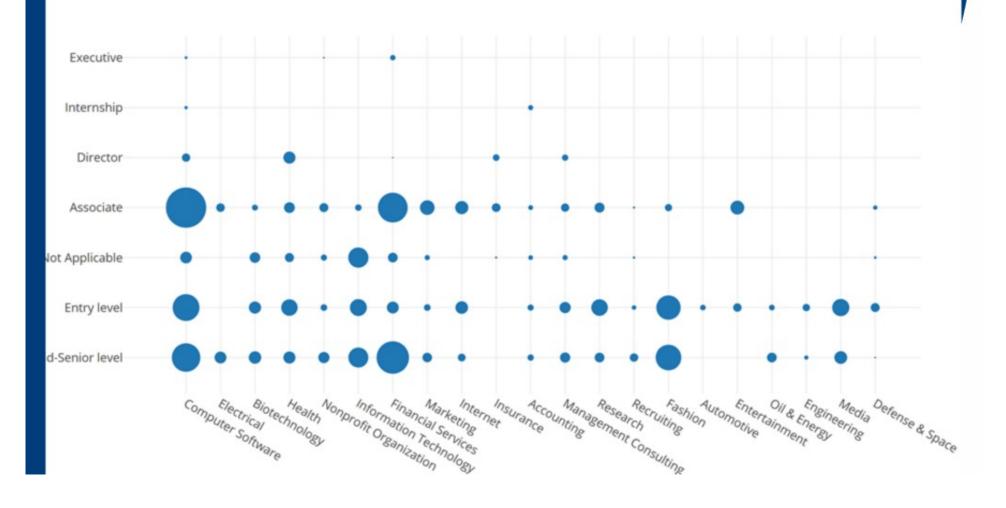




Industry vs Seniority Level (Emphasized by Applicants)









Algorithm

- Feature: Skills & Qualifications
- Label: Conversion rate
 Conversion Rate =
 Number of Applicants/Number of Views

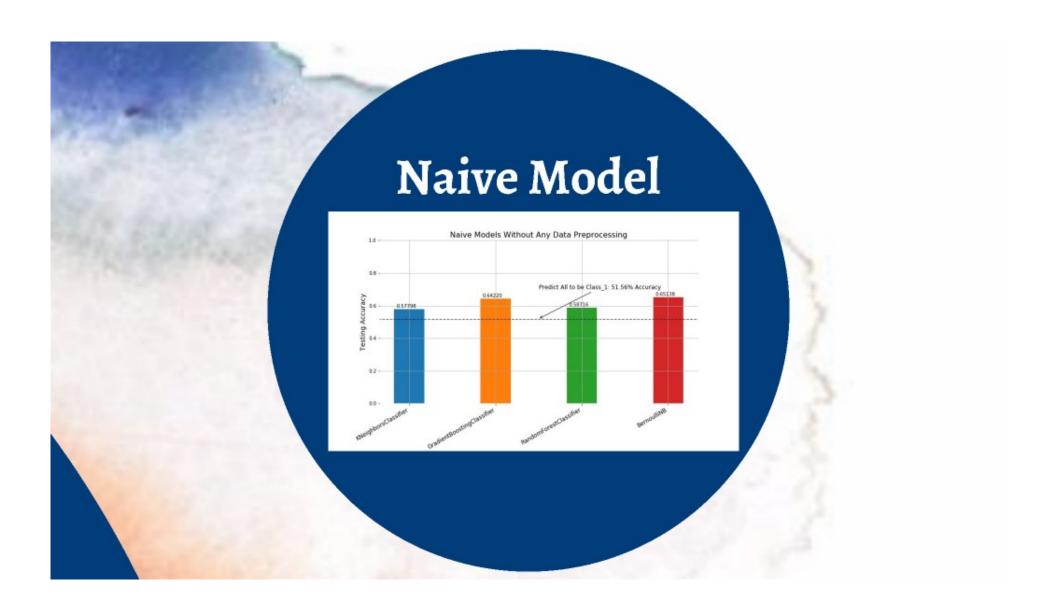
The median of conversion rate is **0.2**, which means every **5** views has **1** applicants

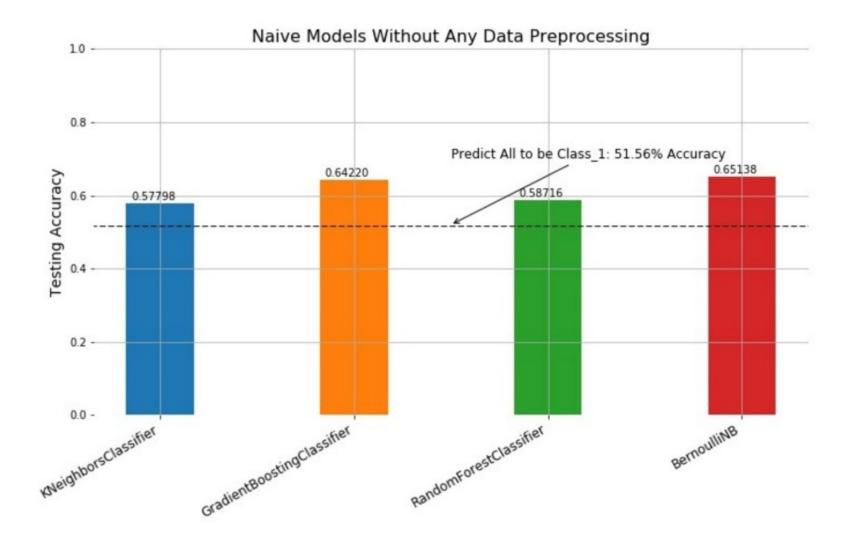
- label '1': conversion rate > **0.2** (good)
- label '0': conversion rate <= 0.2 (limited)

Naive Model

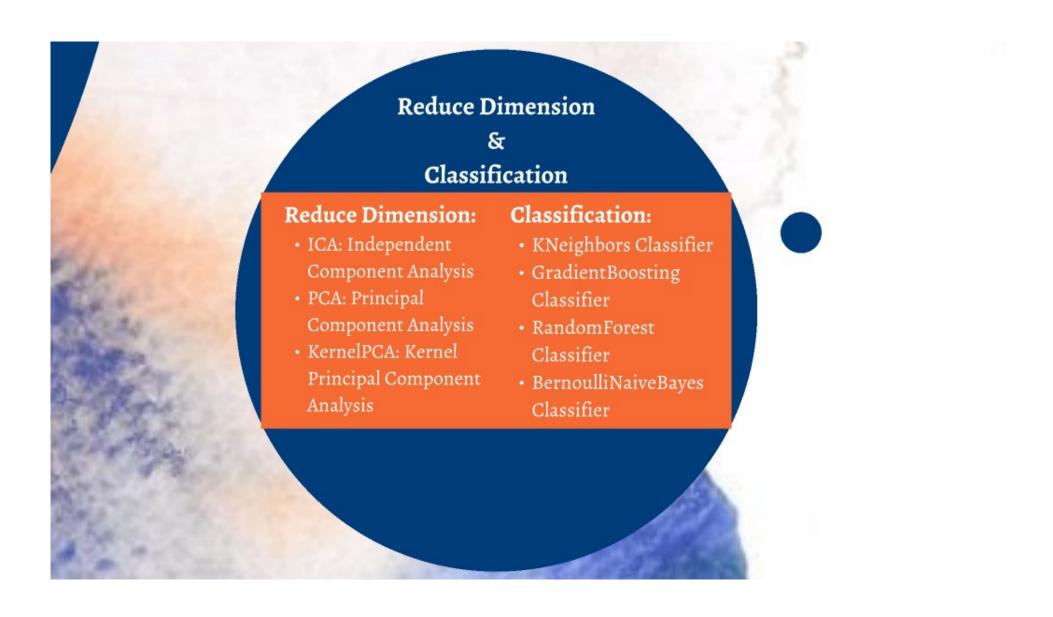
> Data Cleaning & Transforming

Reduce
Dimension
&
Classification

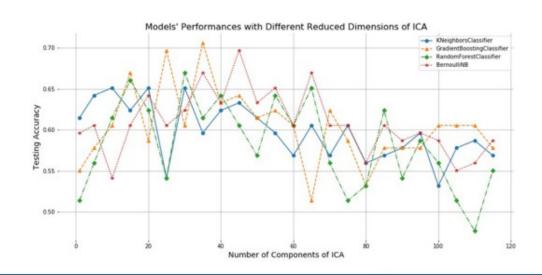




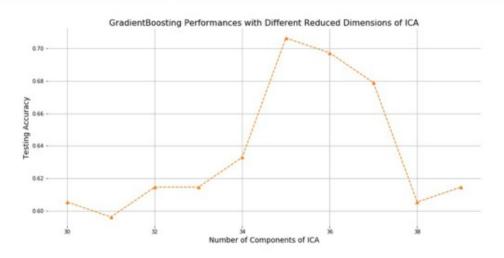


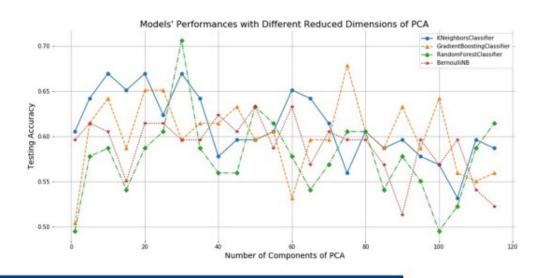


```
def train after DR(learners, dr, X train tfidf, X test tfidf, y train, y test):
    function to trian machine learning algorithms after dimension reduction method
    'learners': list of machine learning algorithms
    'dr': dimension reduction method
    'X train tfidf, X test tfidf, y train, y test': training data and testing data
   result = {}
   # convert sparse matrix to ndarray
   X train = X train tfidf.toarray()
   X test = X test tfidf.toarray()
   # fit Dimension Reduction algorithm
   reduced X train = dr.fit transform(X train)
   reduced X test = dr.transform(X test)
   # fit machine learning algorithms
    for learner in learners:
        learner.fit(reduced X train,y train)
       pred = learner.predict(reduced_X_test)
        acc = np.mean(pred == y test)
       result[learner. class . name ] = acc
    return result
```



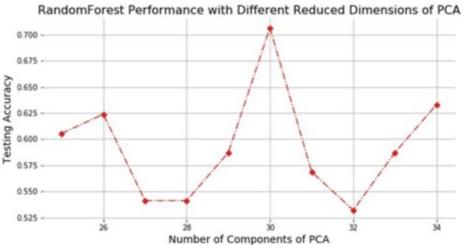
Gradient Boosting Classifier has the best performance based on first 35 ICs testing accuracy around 71%

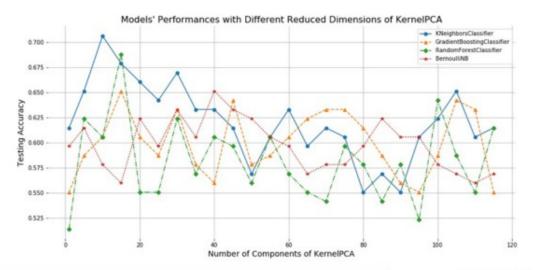






Random Forest Classifier has the best performance based on first 30 PCs: testing accuracy around 71%

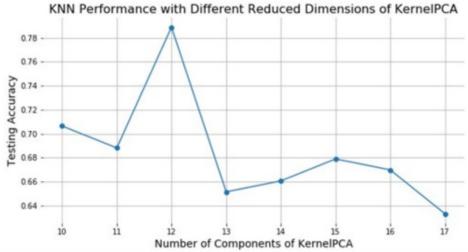




Best! 79% Data: first 12 PCs with Kernel PCA

Model: KNeighbors Classifier

KNeighborsClassifier has the best performance based on first 12 PCs: testing accuracy around 79%



```
{'PC0': [u'experi', u'learn', u'machin', u'statist', u'data'],
'PC1': [u'data', u'abil', u'year', u'model', u'busi'],
'PC10': [u'knowledg', u'least', u'minimum', u'advanc', u'year'],
'PC2': [u'year', u'learn', u'degre', u'analyt', u'experi'],
'PC3': [u'data', u'year', u'least', u'experi', u'strong'],
'PC4': [u'data', u'experi', u'viewpoint', u'plu', u'mathemat'],
'PC5': [u'least', u'busi', u'abil', u'statist', u'quantit'],
'PC6': [u'requir', u'knowledg', u'minimum', u'busi', u'model'],
'PC7': [u'data', u'plu', u'least', u'big', u'languag'],
'PC8': [u'languag', u'common', u'data', u'experi', u'program'],
'PC9': [u'experi', u'softwar', u'requir', u'least', u'analyt']}
```

	precision	recall	f1-score	support
Unpopular_Post Popular Post	0.84	0.70 0.88	0.76 0.81	53 56
avg / total	0.80	0.79	0.79	109





Online Utility - English Language - Text - Math - Other -

Text Analyzer

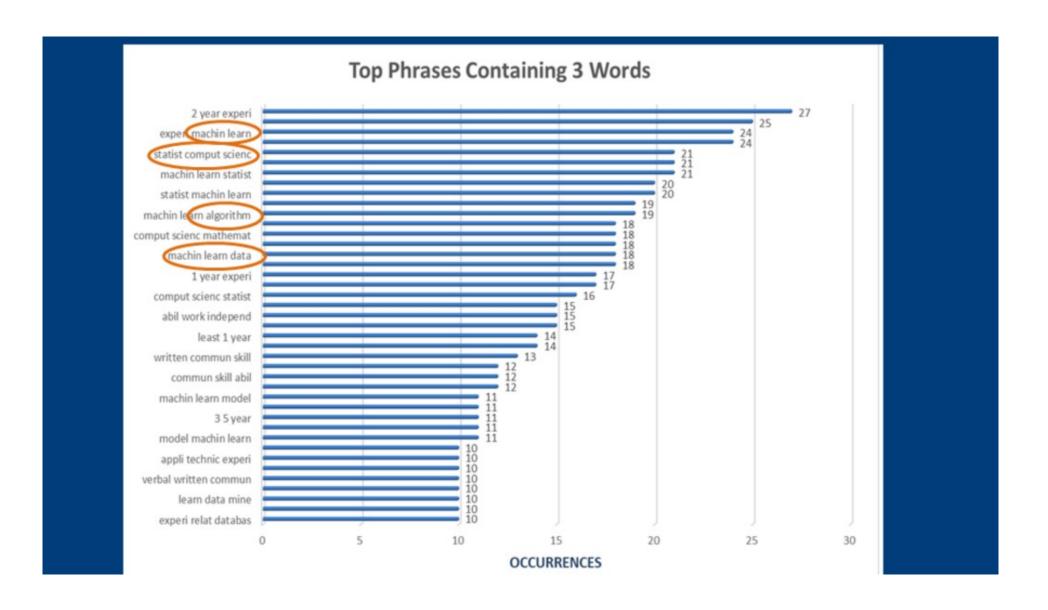
d Like 217

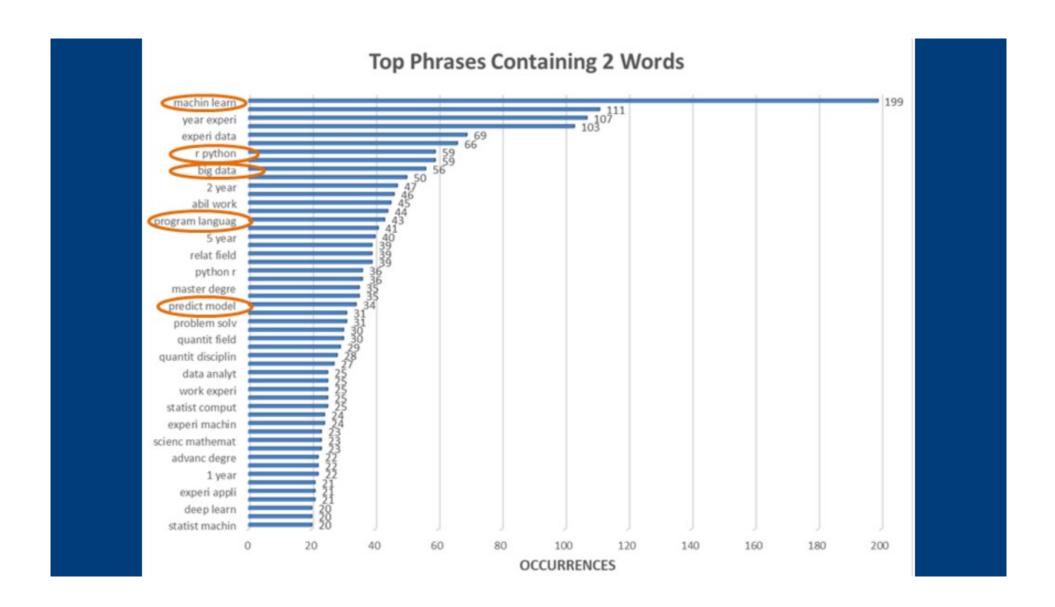
y Tend

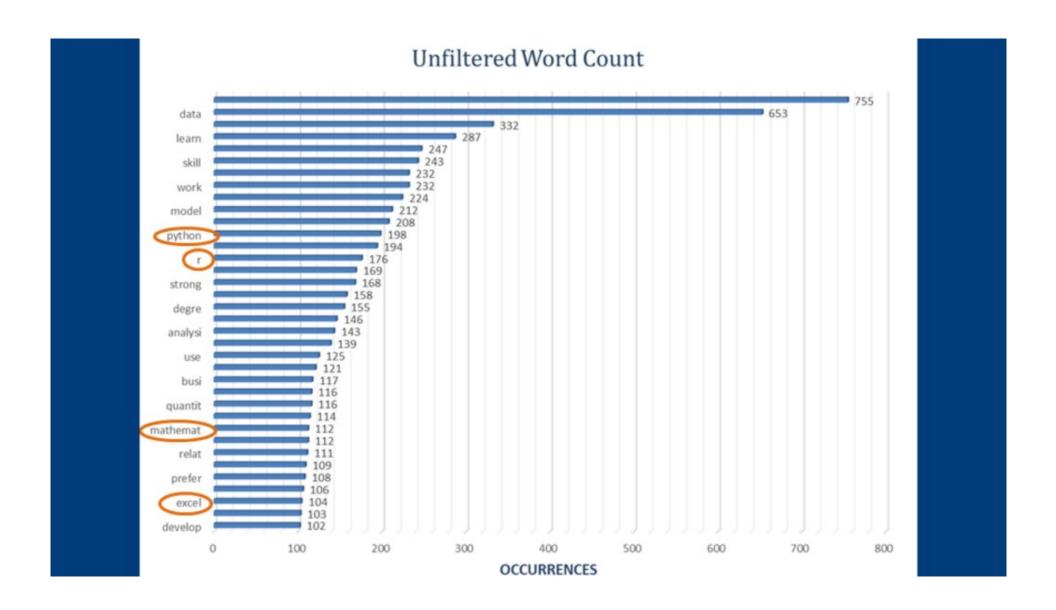
Free software utility which allows you to find the most frequent phrases and frequencies of words. Non-English language texts are supported. It also counts number of words, characters, sentences and syllables. Also calculates lexical density.

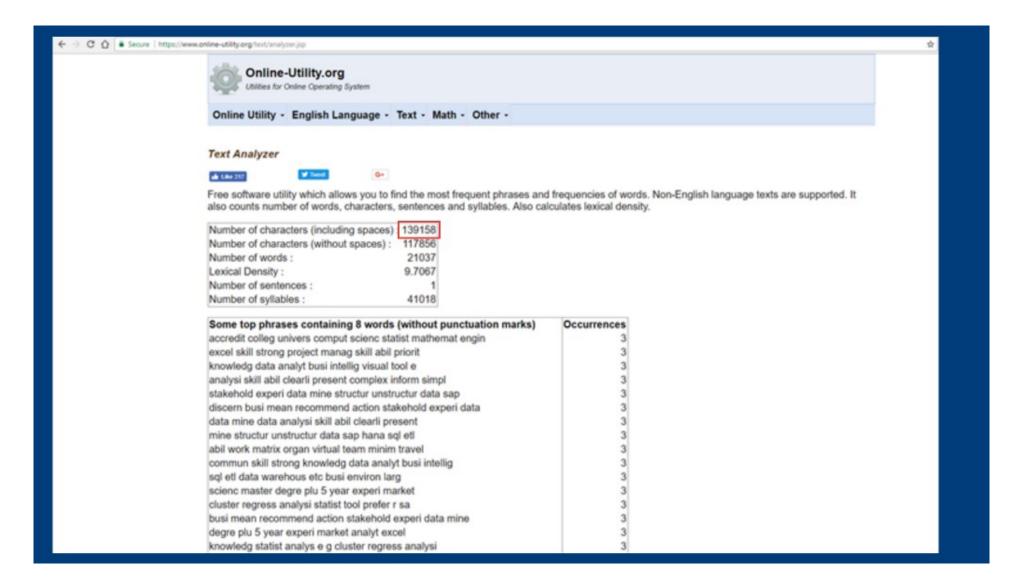
Number of characters (including spaces): 126926
Number of characters (without spaces): 107381
Number of words: 19264
Lexical Density: 9.9408
Number of sentences: 1
Number of syllables: 36985

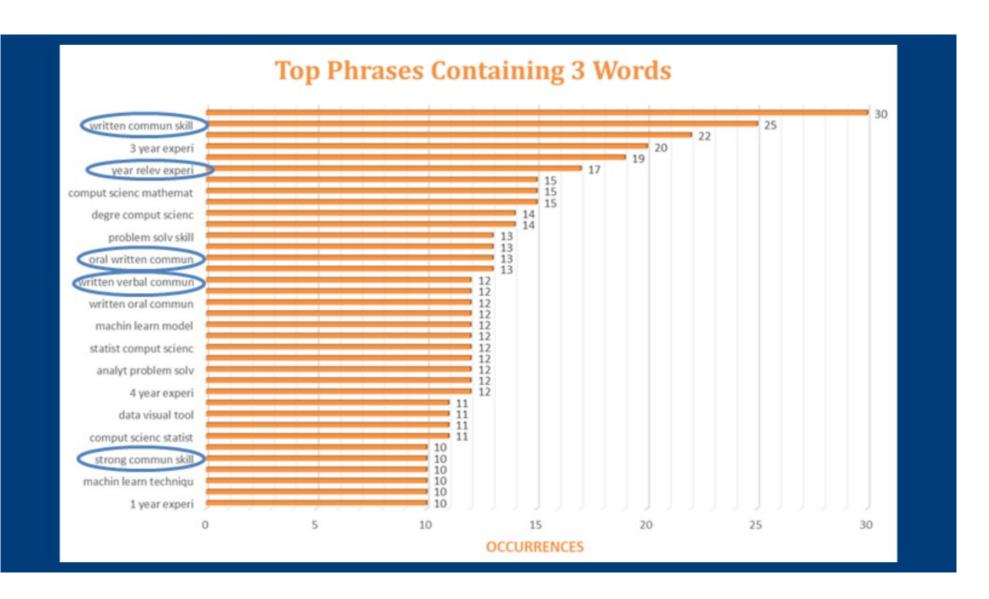
Some top phrases containing 8 words (without punctuation marks)	Occurrences
accredit colleg univers comput scienc statist mathemat engin	4
least 1 year experi open sourc program languag	4
least 1 year experi machin learn least 1	4
experi machin learn least 1 year experi relat	4
experi open sourc program languag larg scale data	4
techniqu cluster classif regress decis tree neural net	4
year experi machin learn least 1 year experi	4
year experi open sourc program languag larg scale	4
machin learn least 1 year experi relat databas	4
manipul analyz complex high volum high dimension data	4
1 year experi machin learn least 1 year	4
cluster classif regress decis tree neural net support	4
open sourc program languag larg scale data analysi	4
1 year experi open sourc program languag larg	4
regress decis tree neural net support vector machin	4
analyz complex high volum high dimension data vari	4

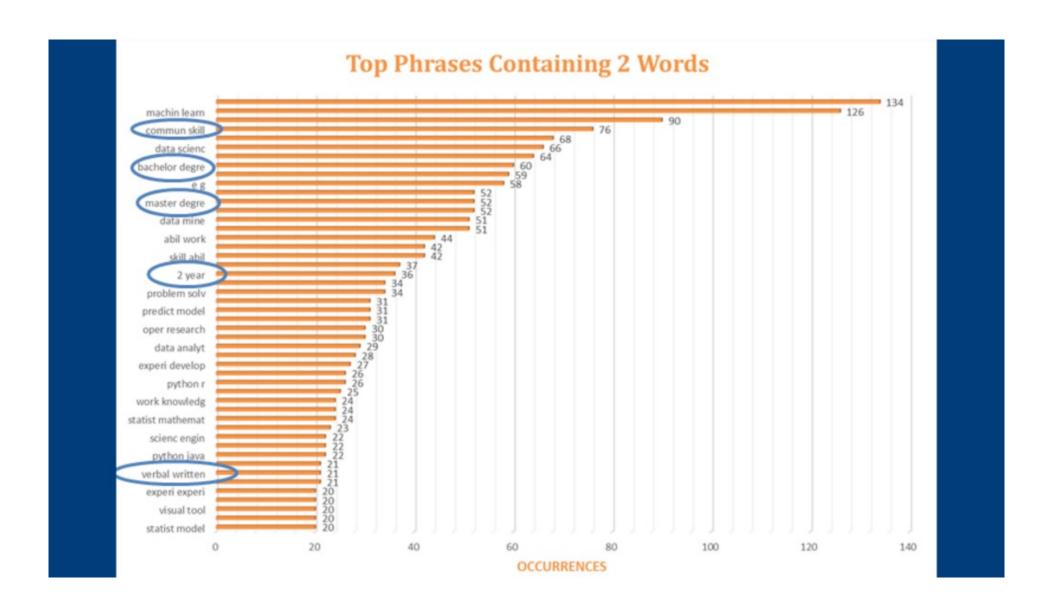


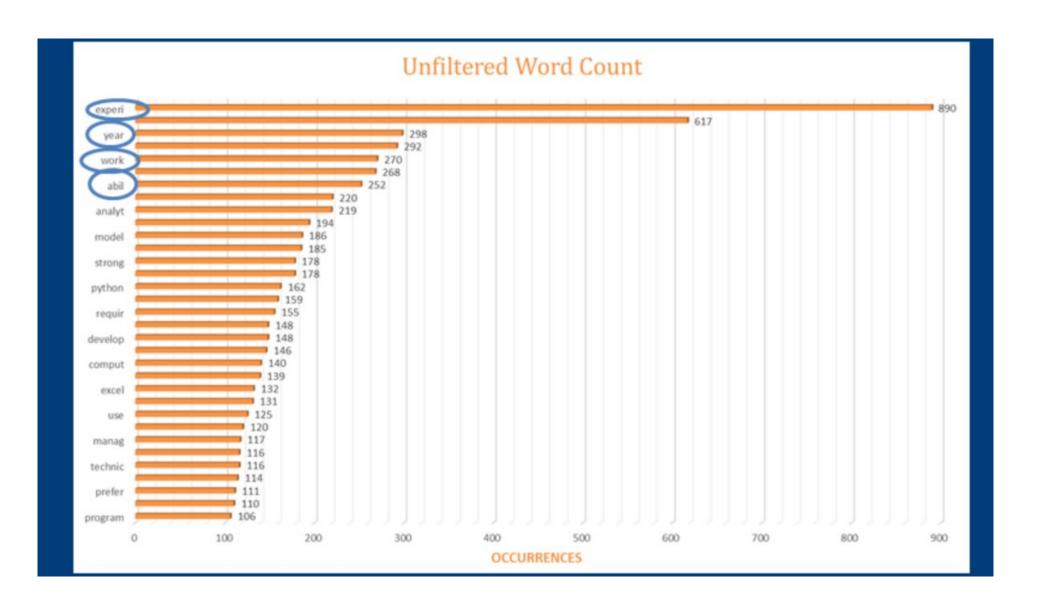












Suggestion & Model

- Contain more specific & professional skills
- More consistent and pertinent to the Industry
- Try to avoid too general words, e.g. oral, written, ability, work
- Be gentle on the experience requirements
- Model to predict whether a job posting is popular ot not based on the skills and qualification description





