CSE 4990/6990 – Big Data and Data Science

Predicting PlayStore Rating for Apps

Team: 5DBMinds

Ayush Raj Aryal

Lucas Andrade Ribeiro

Naila Bushra

Naresh Adhikari

Project Overview

- Predicting Google PlayStore Ratings for new applications
- Used data of a large number of existing applications
- Selected App attributes based on their influence on the ratings

Data Collection

- The data of this project has been collected from the author of the github repository GooglePlayAppsCrawler
- GitHub Link: https://github.com/MarcelloLins/GooglePlayAppsCrawler.git

Data Collection


```
"AppSize":-1.0,
"Category":"SOCIAL",
"ContentRating":"Rated 12+",
"CoverImgUrl":"https://lh3.googleusercontent.com/ZZPdzvlpK9r_Df9C3M7j1rNRi7hhHRvPhlklJ3lfi5jk86Jd1s0Y5wcQ1QgbVaAP5Q=
"CurrentVersion":"Varies with device",
"Description":"Keeping up with friends is faster than ever.• See what friends are up to• Share updates, photos and v
"Developer":"Facebook",
"DeveloperEmail":"android-support@fb.com",
"DeveloperNormalizedDomain":null,
"DeveloperPrivacyPolicy":"https://www.facebook.com/about/privacy/\u0026sa=D\u0026usg=AFQjCNGsgQ5qAO5ohRTZIcLRwgVSGnk
"DeveloperURL":"/store/apps/developer?id=Facebook",
"DeveloperWebsite":"facebook.com",
"HaveInAppPurchases":false,
"Instalations":"1,000,000,000 - 5,000,000,000",
"IsFree":true,
"IsTopDeveloper":true,
"LastUpdateDate":{ ⊞ },
"MinimumOSVersion":"Varies with device",
"Name":"Facebook",
"PhysicalAddress":"",
"Price":0.0,
"PublicationDate":{ ⊞ },
"ReferenceDate":{ ⊞ },
"RelatedUrls":[ ⊞ ],
"Reviewers":-1.0,
"Reviews":[ ⊞ ],
"ReviewsStatus":"Visited",
"Score":{ ⊟
    "Count":3.0383292e+07,
    "FiveStars":0.0,
    "FourStars":0.0,
    "OneStars":0.0,
    "ThreeStars":0.0,
    "Total":3.99822998046875e+14,
```

Data Preprocessing

- Finalized attributes
  - AppSize
  - Price
  - IsTopDeveloper
  - HaveInAppPurchase
  - IsFree
  - PublicationDate
  - LastUpdateDate
  - Installations
  - Category
  - Developer
  - Name
  - ContentRating
  - Description

Data Transformation

- AppSize, Price
  - Numerical Attributes
- IsTopDeveloper, HaveInAppPurchase, IsFree
  - Have True and False values
- PublicationDate, LastUpdateDate, Installations
  - Text to numerical value conversion
- Category, Developer, Name, ContentRating, Description
  - Requires text vectorization

Document Vectorization

- Why Document Vectorization?
  - Machine Learning Algorithms are ‘noumbrevorous’
  - Supervised/Unsupervised: Algorithms take inputs, give outputs thus generate a ‘the most general’ mapping of the data.
  - All inputs/outputs == numbers

Document Vectorization

- Vectorization By Example:
  - SMS-1: "Happy Thanks Giving !"
  - SMS-2: " Happy. Thank you. Wish you great Christmas."
- Un-Stemmed Vectorization: Use the word as you find it.

Document	Happy	Thanks	Giving	you	Thank	Wish	christmas	great
Sms-1	1	1	1	0	0	0	0	0
Sms-2	1	0	0	1	1	1	1	1

- Stemmed Vectorization: Remove Basic Words and Use only base or root of a word.

Document	Happy	Thank	Give	wish	christmas	great
Sms-1	1	1	1	0	0	0
Sms-2	1	1	0	1	1	1

Document Vectorization

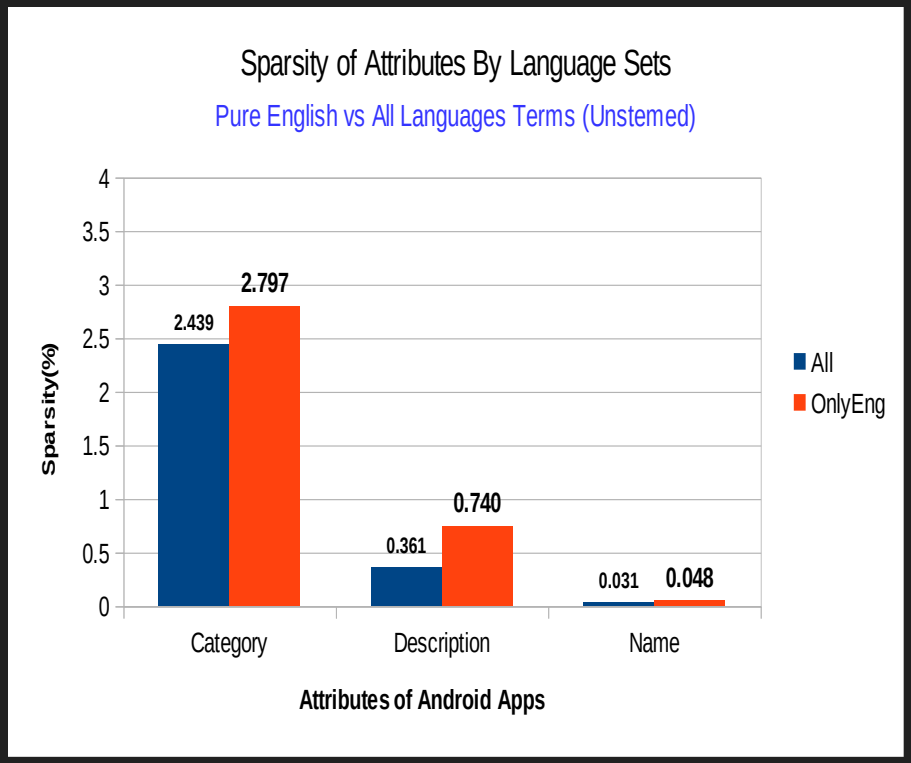
- **STEPS FOR VECTORIZATION:**
- Step-1: Tokenization
  - Generate collection of words fro each documents, Throw Away Redundant Words or punctuation marks: ‘you’, ‘!’
- Step-2: Counting
  - Find Frequency of occurrences of a word/token in a document
- Step-3: Normalization
  - Divide vectors SMS-1 and SMS-2 by |SMS-1| and |SMS-2|

Document Vectorization

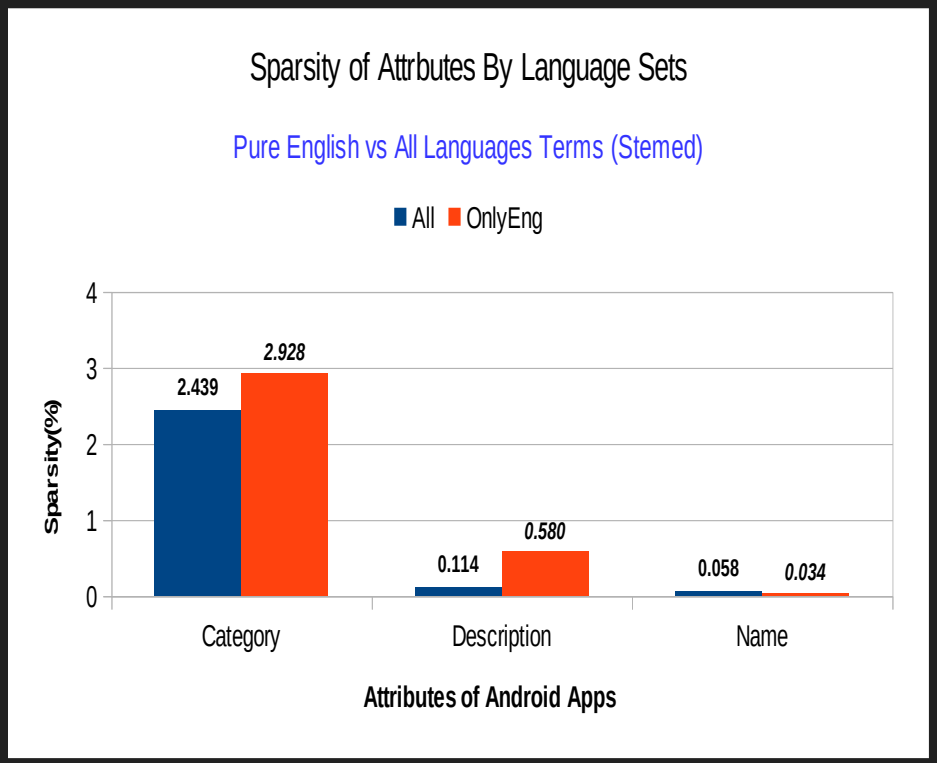
- Apps' Attributes Vectorized:
  1. **Category** : 41 Categories, monolingual corpus
  2. **Description**: One app one description. Multilingual corpus
  3. **Name**: one app one name, multilingual corpus
  4. **Content Rating**
  5. **Developer**

Document Vectorization

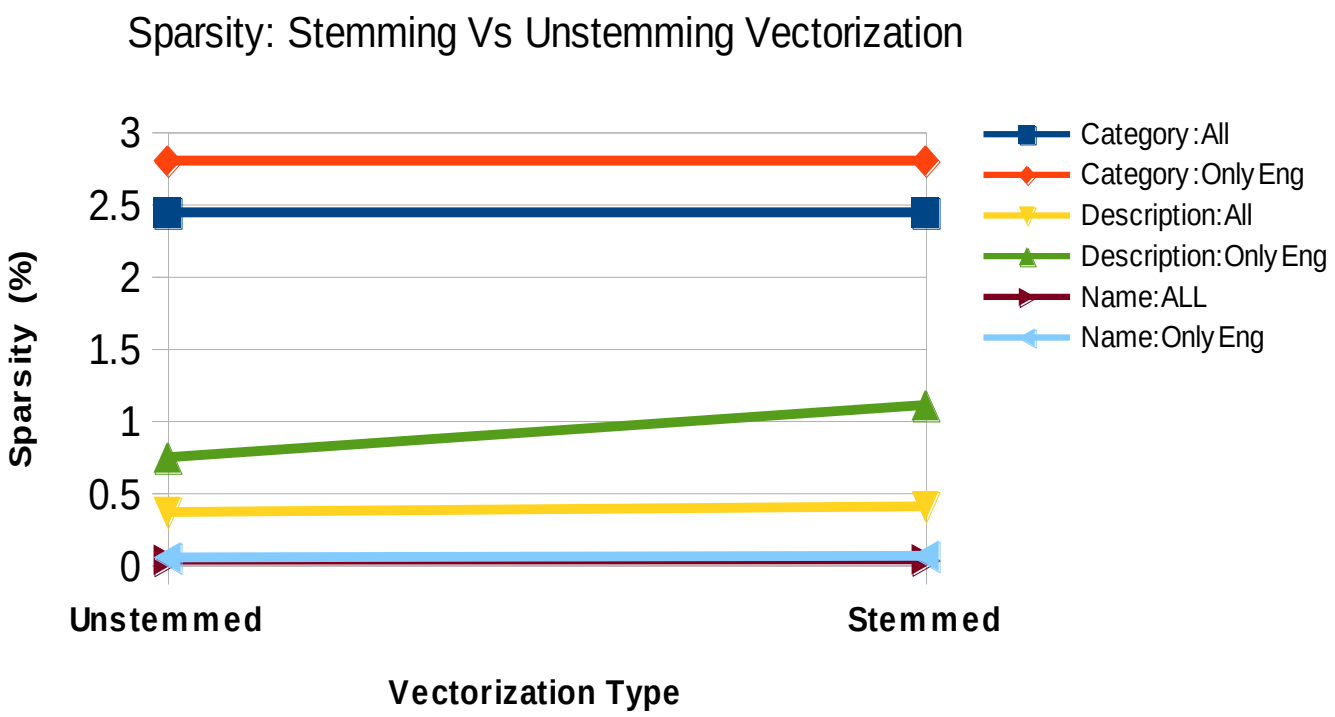
Sparsity: Unstemmed Vectorization



Sparsity: Stemmed Vectorization



Document Vectorization

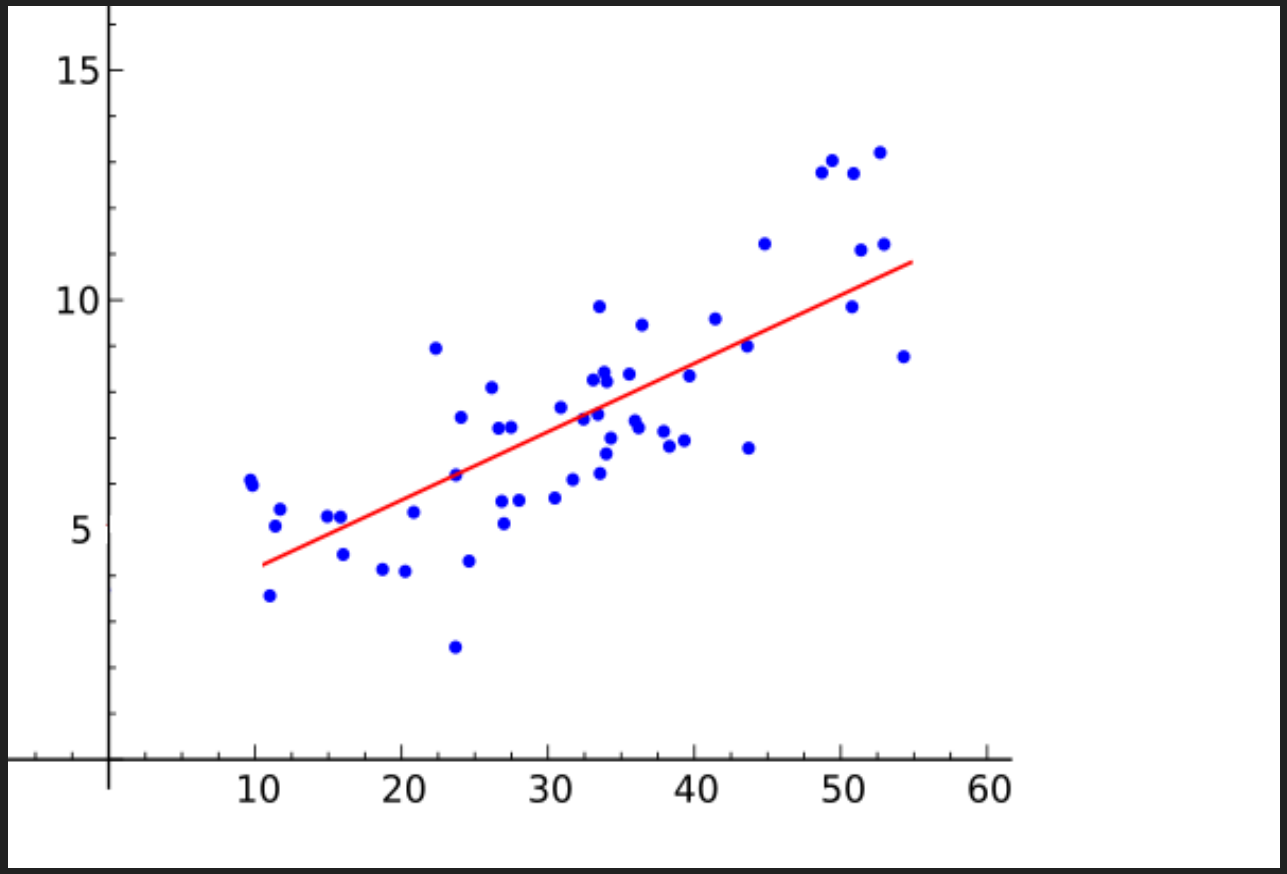


Linear Regression

- A simple regression model for supervised learning
- Used to predict a target variable Y which is linearly dependent on other independent variable(s) X
- Given the independent variables X1, X2,…,XN for k data points the model can be represented as

$$Y_k = \beta_0 + \beta_1 X_{k1} + \ldots + \beta_N X_{kN} + \varepsilon$$

Linear Regression



Linear Regression

- Residual sum of squares (RSS), *e* = *y*-*y'*

$$\mathrm{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$$

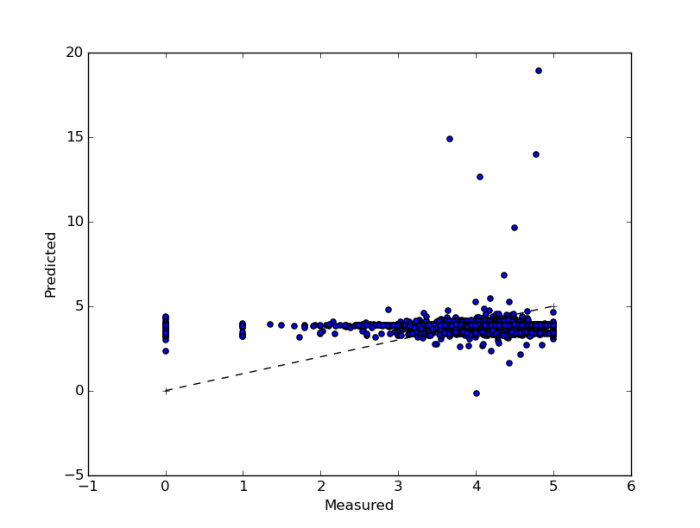
- Returns the coefficient of determination R^2 of the prediction
  - The coefficient R^2 is defined as (1 - u/v)
  - u = RSS
  - V = residual sum of squares
    - ((y_true - y_true.mean()) ** 2).sum()

Linear Regression

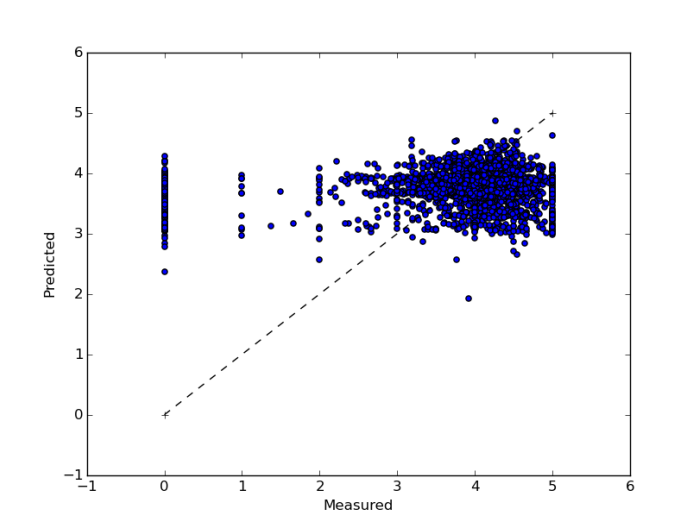
- Used cross validated prediction to visualize prediction errors
- K-fold cross validation (k=4) technique was applied to randomize the train and test dataset
- Vectorization resulted in high dimensional feature set
- Principal Component Analysis was done to reduce the number of features to around 100

Linear Regression

All features

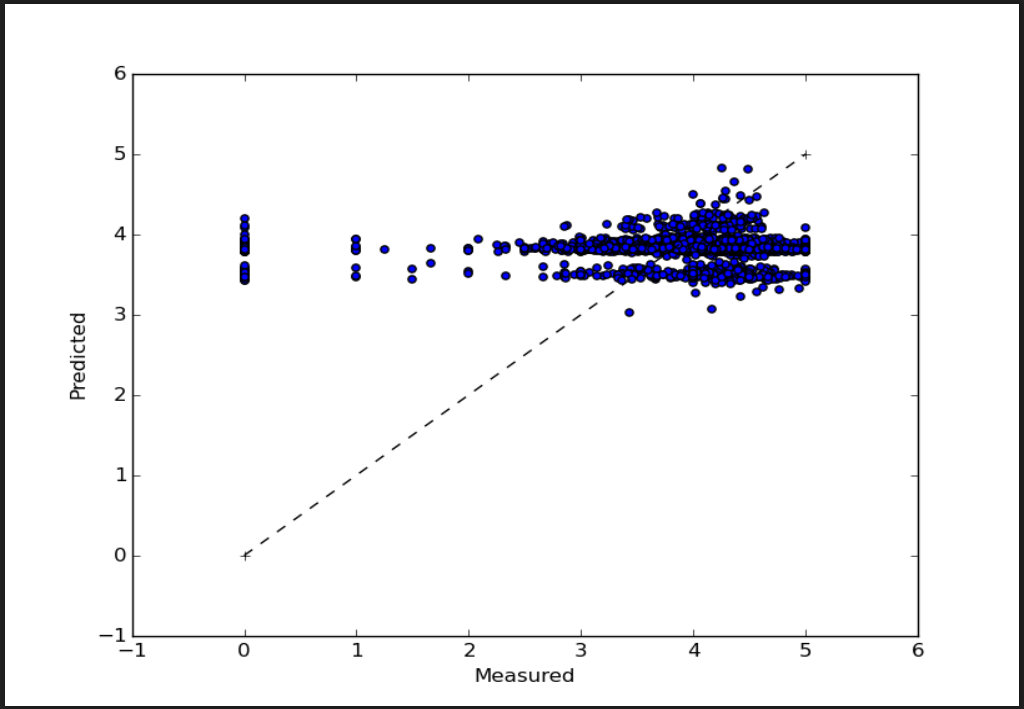


Without higher dimensional features



Linear Regression

Without vectorized features



Linear Regression

All features

4 fold cross validation	iter1	iter2	iter3	iter4
Residual sum of squares	1.27	1.38	1.36	1.10
Variance score	0.56	0.31	0.40	0.23

Without higher dimensional features

4 fold cross validation	iter1	iter2	iter3	iter4
Residual sum of squares	1.36	1.33	1.18	1.20
Variance score	0.49	0.38	0.19	0.25

Without vectorized features

4 fold cross validation	iter1	iter2	iter3	iter4
Residual sum of squares	1.10	1.27	1.30	1.33
Variance score	0.37	0.45	0.12	0.24

Classification

- Classification is used to identify the category of a new observation on the basis of training data whose categories are know.
- We used transformed numerical and boolean features and the vectorized textual columns to predict the rating scores for new applications.
- The scores were categorized into classes ranging from 1 to 5.

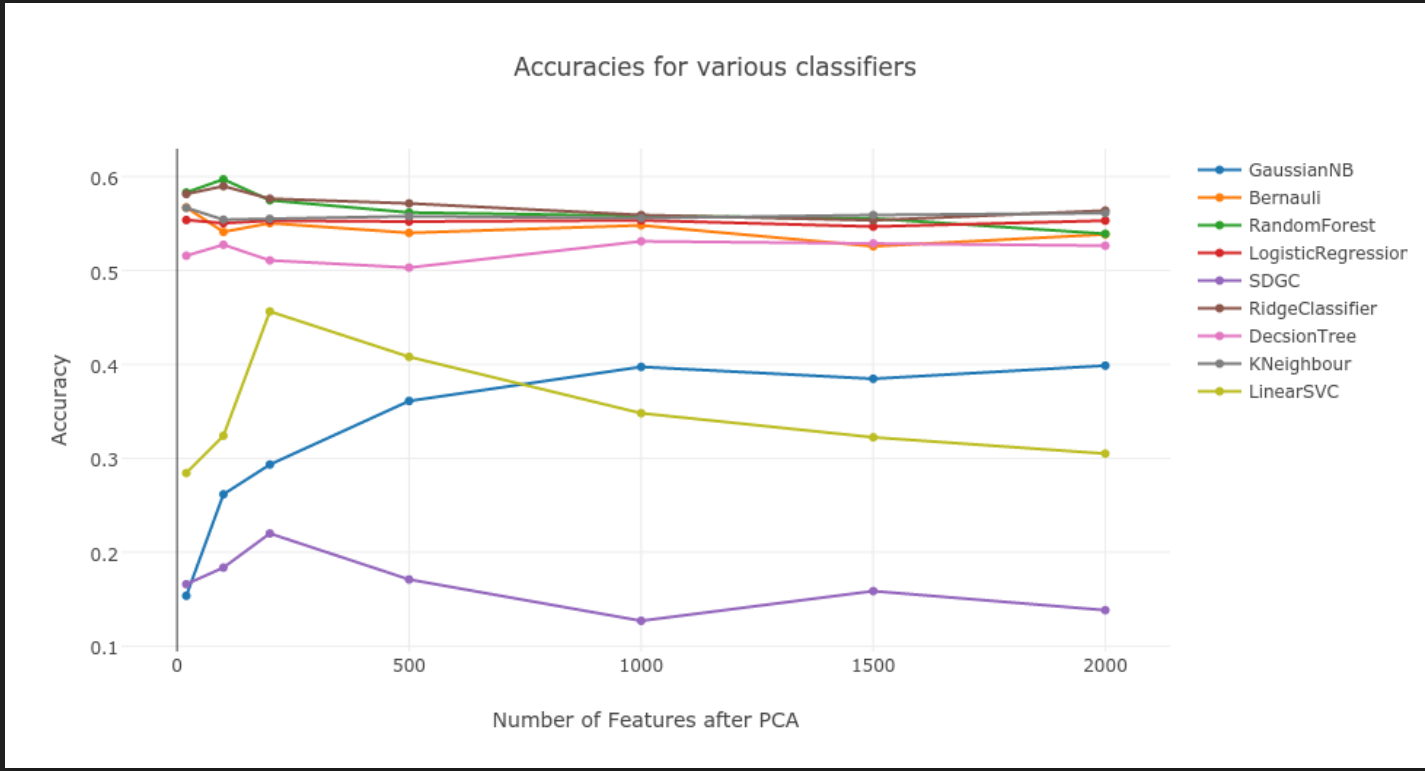
Algorithms for Classification

- Gaussian Naive Bayes
- Bernoulli Naive Bayes
- Random Forest Classifier
- Logistic Regression Classifier
- SGD Classifier
- Ridge Classifier
- Decision Tree Classifier
- K Nearest Neighbors Classifier
- Linear SVC

Methodology for Classification

- The feature set had large number of columns after vectorization. Principal Component Analysis (PCA) was implemented to reduce the dimensions of feature set.
- Then K-fold Cross validation was implemented and accuracy for various classifiers was studied for k=10.
- The dataset is splitted into k consecutive folds. Each of the k folds is used as a validation set while remaining k-1 is used training set.

Comparison of Classifier Performance



Tuning of Classifier

- Selected classifiers with higher accuracies.
- Implemented Grid Search technique to get set of optimal parameters.
- Grid Search creates a grid of all possible parameter combinations and tests the model with all the combinations
- It gives us idea about which set of parameters gives the best results.

Results After Tuning

	Accuracy Before Tuning (%)	Accuracy After Tuning (%)
Random Forest	59.7	69.33
Ridge Classifier	58.9	64.7
KNN Classifier	55.4	62.11
Logistic Regression	55	65.7
Decision Tree	52	60
	Number of Components	*100*
	Data Points	*3000*

- Best Results: Random Forest ~ 70% accurate

Issues

- The hardware infrastructure we have was not enough for vectorizing the whole dataset of 1.2 Million applications
- Used only 3000 data points in order to limit the computational time and memory usage.
- The vectorizer was filtered to use only the English words to avoid memory issues.

Scope of improvements

- Use parallel processing to handle all available data
- Use correlation between features to find out most usable features

BigData

Predicting PlayStore Rating for Apps

Overview

What is the project

The project is a software that uses BigData and Data Science techniques to predict "Google PlayStore" ratings for new applications using patterns found in a file that contains all ratings data from applications that are already available in the "Google Playstore".

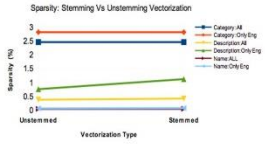
Data

Data transformation

- The data of this project has been collected from the author of the GitHub repository GooglePlayAppsCrawler.
- GitHub link: https://github.com/MarcelloLins/GooglePlayAppsCrawler.git
- Attributes used : AppSize, Price, IsTopDeveloper, HaveInAppPurchase, isFree, PublicationDate, LastUpdateDate, Installations, Category, Developer, Name, ContentRating, and Description.
- Some of the attributes needed vectorization while others were numerical or could be transformed.
- The attributes that needed vectorization are: Name, Description. Category, Content Rating, Developer.

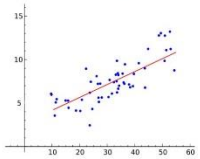
Work

Document Vectorization



- The attributes Category, Developer, Name, ContentRating, and Description, were transformed into numerical information.

Linear Regression

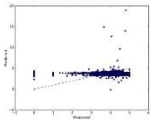


- A simple regression model for supervised learning was used to predict a target variable Y which is linearly dependent on other independent variable(s) X
- Used cross validated prediction to visualize prediction errors.
- K-fold cross validation (k=4) technique was applied to randomize the train and test dataset.
- Principal Component Analysis was done to reduce the number of features to around 100.

More Work

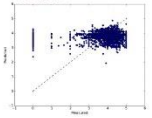
Linear Regression 2

With all features



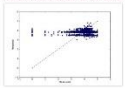
10 fold cross validation	Iter1	Iter2	Iter3	Iter4
Residual sum of squares	1.27	1.38	1.36	1.10
Variance score	0.07	0.13	0.04	0.01

Without high dimensional features



4 fold cross validation	Iter1	Iter2	Iter3	Iter 4
Residual sum of squares	1.36	1.33	1.18	1.20
Variance score	0.49	0.38	0.19	0.25

Without vectorized features



Classification

- Classification is used to identify the category of a new observation on the basis of training data whose categories are know.
- Selected classifiers with higher accuracies.
- Implemented Grid Search technique to get set of optimal parameters.

	Accuracy Before Tuning (%)	Accuracy After Tuning (%)
Random Forest	59.7	69.33
Ridge Classifier	58.9	64.7
KNN Classifier	55.4	62.11
Logistic Regression	55	65.7
Decision Tree	52	60
	Number of Components	*100*
	Data Points	*3000*

Members

Faculty

- Dr. Somya Mohanty

Students

- Ayush Raj Aryval
- Lucas Ribeiro
- Naila Bushra
- Naresh Adhikari

Thank you