Introduction

The world of books is compelling to people of all ages. Whether you are heading for vacations, spending some free time on your own or just enjoy reading a good work of literature, reading will ensure you are having some quality time. What makes a book attractive and interesting? This work will focus on exploring the features which are important to forecast a good rating for a book.

Publishing companies spend a lot of time and resources to search (filter out) literature candidates that attract a broad audience. A data driven approach to this problem would help the publisher make decisions.

Based on 5-star ratings, the task is to predict the ratings of the books from different data sources.

1 Collecting the data

This research starts with data collection. Three data sources were considered:

- Open Library dumps. The data from the dumps comes in the following format:

  - **type** of record (/type/edition, /type/work etc.)
  - **key** unique key of the record. (/books/OL1M etc.)
  - **revision** revision number of the record
  - **last_modified** last modified timestamp
  - **JSON** the complete record in JSON format

  For this project we are focused on the last record which comes in JSON format.

- Amazon website. Based on the Open Library dumps there was an attempt to scrape the book features from Amazon website using book ISBN identifiers. These attempts were not successful as per limitations from Amazon to suppress web scraping, especially from a given IP address. There is indeed a collection of datasets from Amazon (http://jmcauley.ucsd.edu/data/amazon/). The features extracted from this database are mostly fit for building a recommendation system, which is not the part of this work.

- Goodreads This web resource complements Open Library dumps. It provides an API to retrieve book details. More details on Goodread API here.

As a result, in this work we use two data sources: Open Library dumps and Goodreads.

2 Data cleaning, data processing, data wrangling

In this work we start with data retrieved from Open Library dumps. As per the previous section, the data come in a JSON format. The data in json format should be transformed to a data table. In this project we are using pandas python package.

First we read the dump using the command:

```
 olde = pd.read_csv(path_to_dump, sep='\t', header=None, names
   =['Book'], usecols=[4])
```

This will result in a table with one column containing the json. The dump is huge and takes more than 16 GB of RAM. The dump was split in 5 files containing around 29602272 records each.

The transformation is applied in chunks of size 1000. We use Dewey classification to filter out all non-literature works. Will retain only those records starting with 8́ór a letter. For each record we apply json_normalize function from 'pandas.io.json' module. The tranformation is not time efficient and going through all the records was taking a huge amount of time. Had to restrict the number of records for this project. There were chosen first 4000 literature works from each file.

The resulting table contains 82 columns. Columns that contain identification information are discarded:

```
BookInfo.drop([item for l in filter(None, BookInfo.columns.str.
    findall('identi.*')) for item in l], axis=1, inplace=True)
```

Only ISBN will serve as a book identifier, although the same work from the same author could have several ISBNs. ISBN is not only the literature work identifier but also it is linked to the publisher. The ISBN is necessary to retrieve rating information from Goodreads (https://www.goodreads.com).

Goodreads provides API to retrieve the ratings from ISBNs. The ISBNs were parsed from the original table:

```
onlyISBN = booksDF[~booksDF.isbn_10.isnull()]
splittedISBN = onlyISBN.isbn_10.str.findall("\d*[a-z]*")
splittedISBN=splittedISBN.sum()
allISBNs=list(filter(None, splittedISBN))
```

The ratings table is built from Goodreads ratings. The ratings were retrieved in chunks of 500 and json_normalize(d) to build a pandas dataframe.

This table contains 11 columns. The most important is **average_rating**. We will use this value as the dependent variable.

The isbns are provided as strings in the following format: "['isbn1', 'isbn2']". Need to get them to a list object to manipulate:

```
BookInfo['isbn10'] = BookInfo.isbn_10.apply(lambda x: eval(str(x
    )) if type(x)==str else '')
```

To complete the book table it is necessary to add another column *'average_rating'*:

```
BookInfo['average_rating'] = BookInfo.isbn10.apply(lambda x:
    BookRatings[BookRatings.isbn10.isin(x if type(x)=='list' else
     list(x))].average_rating.mean()
```

Not all the books that we have retrieved do have a rating in Goodreads. We will remove the records without an average rating:

```
BkInfNNullRtng = BookInfo[BookInfo.average_rating.notnull()]
```

After a quick look at the histogram (Figure 1):

```
BkInfNNullRtng.average_rating.hist()
```

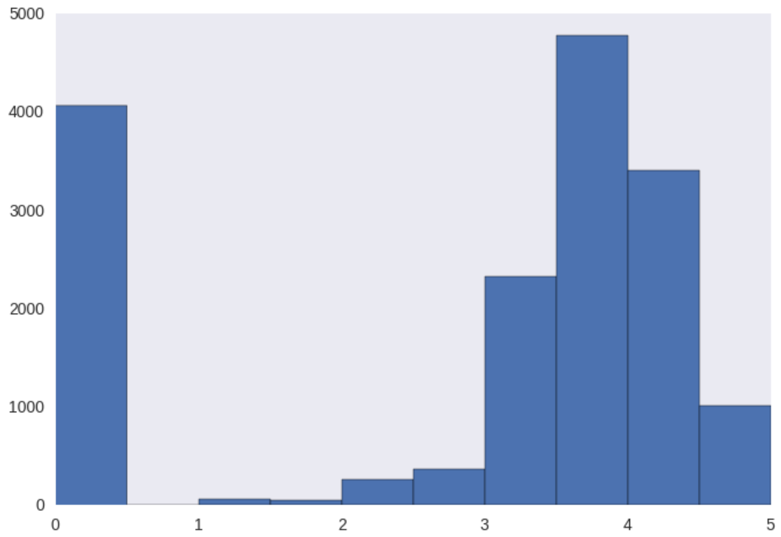

Figure 1: Average ratings distribution plot

It seems that there are a lot of 0 rated books. These books are not rated yet in Goodreads. Removing them as well:

```
BookInfoAverage = BkInfNNullRtng[BkInfNNullRtng.average_rating
    != 0]

```

```
LangCount =  BookInfoAverage['languages'].value_counts()
LangCGT3 = LangCount[LangCount > 3]
BookInfoAverage =   BookInfoAverage[BookInfoAverage.languages.
    isin(LangCGT3.index)]
```

The date of birth from OL dumps was parsed and normalized as the records were having different formats.

To summarize the data frames:

BookInfo: created from Open Library dump. This dataset has book ids (ISBNs) grouped together.

AuthorInfo: created from Open Library dump. This dataset has links to the books through author IDs.

gr_books: created from Goodreads. This dataset has been collected by scraping data in Goodreads. The books are identified by their ISBN. Even the same book (same title, same author but different ISBN) has a different record.

gr_authors: created from Goodreads. The authors dataframe has more information. It relates the books to all the contributors (author, co-author, illustrators, etc.).

Open Library dump data did not contain any book ratings. The ratings were retrieved from Goodreads resources. The dataset was combined from both OL and Goodreads by using ISBN field. The books that were not rated have been excluded from the data set. The fields of interest are:

- Publish date of a book
- Book genre, which was retrieved from Dewey classification. Dewey classification was also parsed to remove the sub-classification. We retained the 8xx Dewey classification code for this dataset. This code belongs to literature works. The scientific, historical, training literature is omitted.
- Book title
- Language
- Number of pages
- Subject places
- Book format

- Number of contributors. This field is not in the dataset but is built by retrieving the number of links to the author of a book.
- Author
  - Gender
  - Number of works
  - Birth date. This field has a lot of missing data. In OL dataset the format was not consistent. Retrieved the year of birth to have a consistent format along this column.

The data above is limited. There is no data on the work content, or metrics from a literature expert. The success of a book cannot be correlated with metrics that would evaluate the skills of an author. This information could be retrieved by utilizing data such as the number of the literature works written by an author and authors age.

3 Exploratory analysis

We start exploring by plotting the distribution of the books' rating number published over the years (Figure 2):

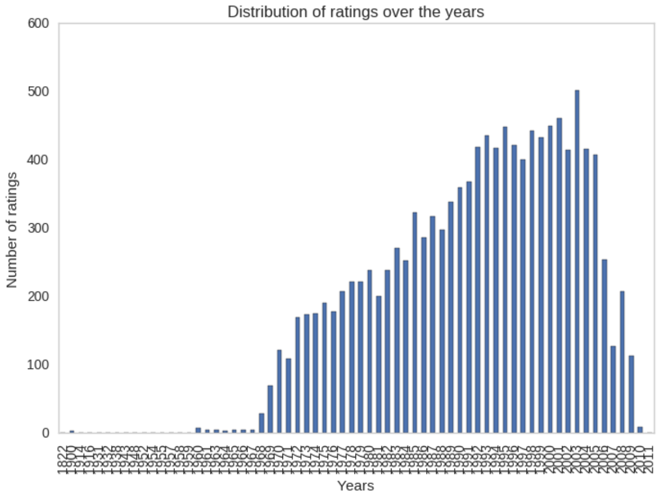

Figure 2: Distribution of book ratings over the years

This is the bar chart of the number of ratings of the books published in a given year. The peak year is 2003. From the dataset that we have this is the publication year for the books rated by most of the people. The same distribution is plotted by months (Figure 3).

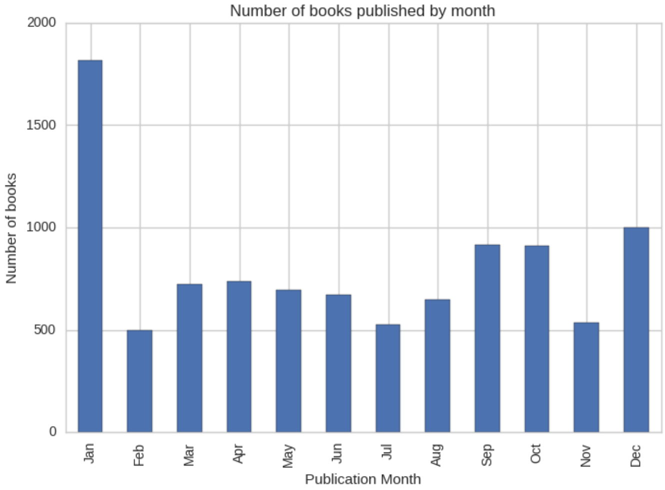

Figure 3: Distribution of ratings per month

In the above plot there is a periodicity on how many books are published per month. The number of published books on February, July and November is lower than other months. January is not really the highest. The number of published books on January is instead an outlier.

The average rating is not dependent on the year, month or day a book is published.

Let's see how the average rating is affected by foreign languages (Figure 4). The size of the dots is weighted by the number of books in each class.

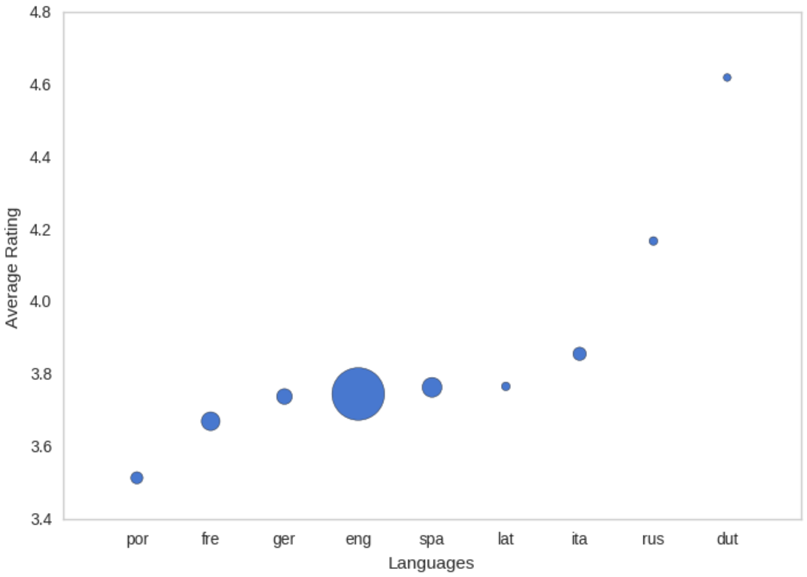

Figure 4: Average ratings by language

Books written in Dutch, Russian, Italian, Latin and Spanish are rated higher than books written in English. On the other hand the number of books written in English dominates over the other language groups. The focus will be in those books written in English.

As mentioned previously we are using Dewey classification to analyze only literature works. The literature works versus others are all the books that have a 8́xxćode. Inside this class there are other subclasses. The distribution of average ratings between different genres of literature is captured in the plot (Figure 5).There is some variability in the distribution of average ratings for each genre. After taking a closer look we see that poetry genre is one of the most highly rated. This could be explained by refined literature skills one requires to write poetry.

The distribution of the books by the number of pages is given by histogram in (Figure 6). The most common book sizes range between 200-500 pages.

We check how the number of pages affects the average rating by plotting the following scatter plot (Figure 7). From the plot we can see that books with number of pages greater than 1̃000 have better ratings. From a quick check the books that have a large number of pages seem to be collections of literature. This is an important observation. Literature collections tend to group the best works together. Still, one should be cautious, because large number of pages and collection are not always correlated together.

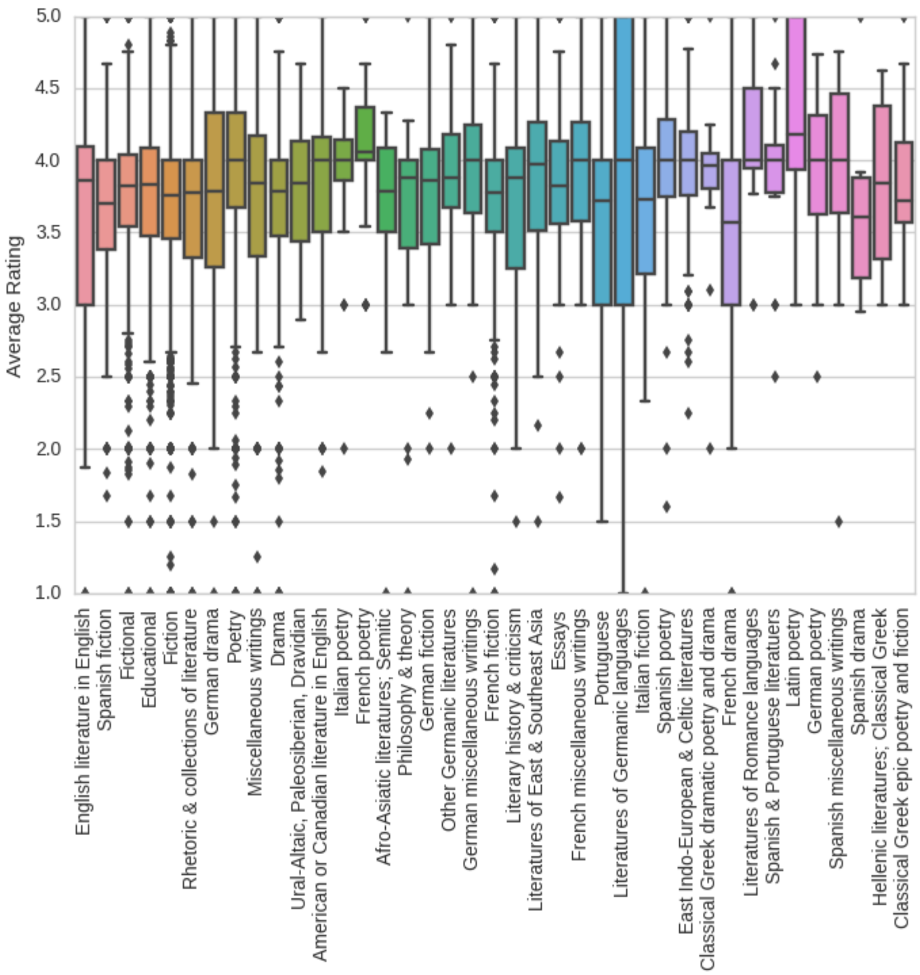

Figure 5: Average ratings by genres

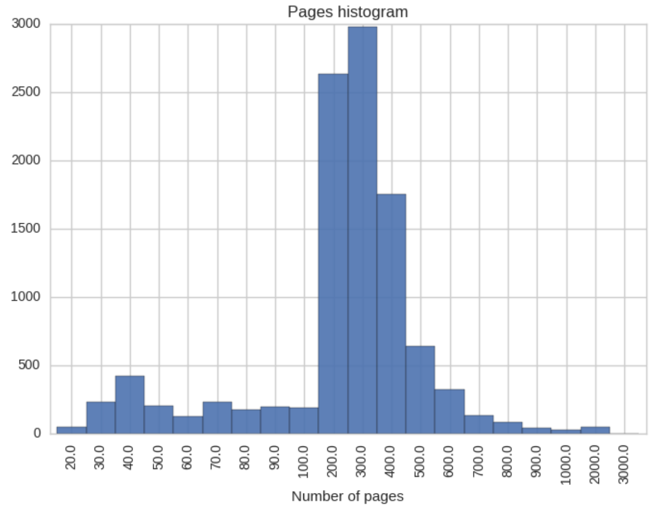

Figure 6: Histogram of books by number of pages

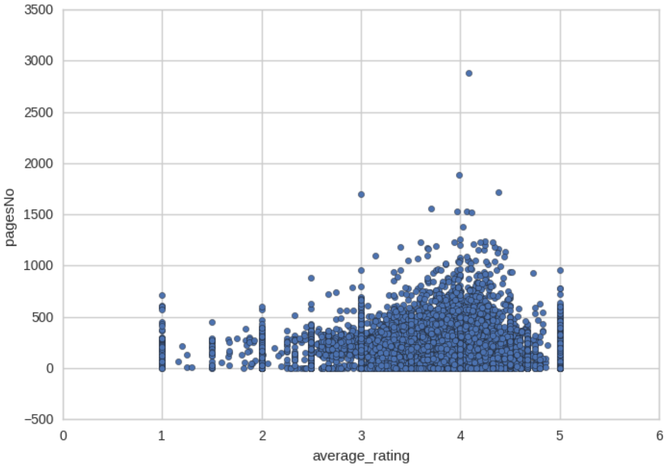

Figure 7: Number of pages vs average ratings.

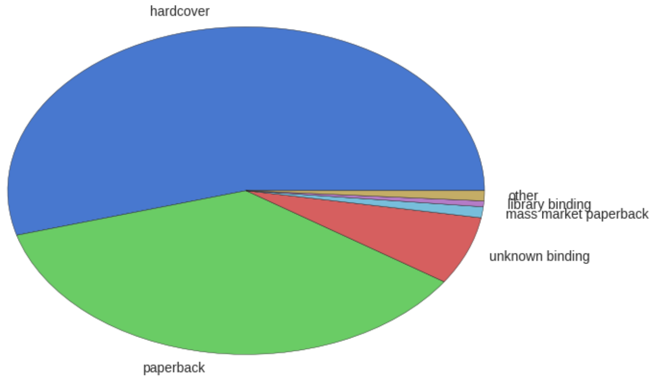

Figure 8: Book formats

From the dataset, we can group the books by their format. The distribution of formats is given in (Figure 8).

To check whether the format of a book affects its rating we look at the following plot (Figure 9).

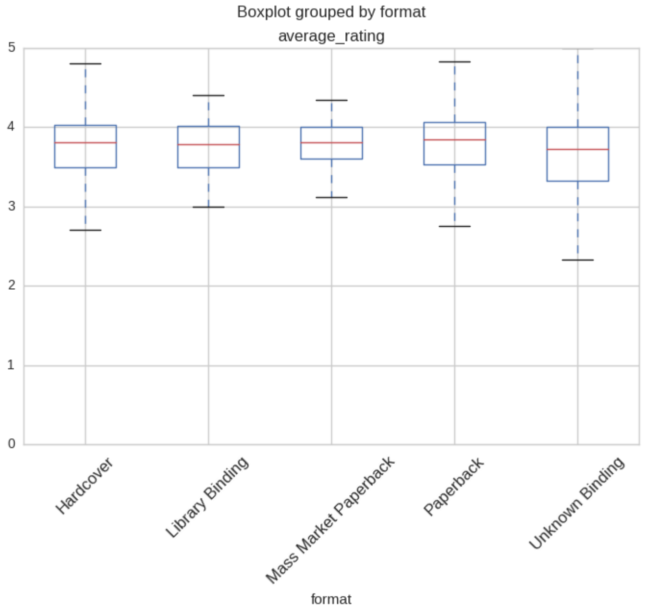

Figure 9: Average rating by book format.

The format does not affect the average rating. The boxplots do overlap on the hinge region. There is no significant difference in the average. The only difference is in the tails of the distributions.

The preliminary analysis gave some insights about the data.

More than half of the rated books are published in 1980-2005 years.

More than half of the books (62%) in our dataset belong to Fiction genre. About 38% of authors in our dataset are females. Women tend to get higher ratings on Drama genre. Most of the books on Educational genre are written by women (61%). However, an educational book written by a man did get higher ratings and from the hypothesis testing this was not a random effect.

Poetry as a genre was the best rated. Writing good poetry is not an easy task and requires good writing skills. A lack of metrics on evaluating poetry and other genres make it difficult to understand the affecting factors.

In this project English language books will be considered. There are other languages that have an average rating higher or lower than English books, but they are not a representative of the population.

4 Feature selection

From the dataset and analysis made so far we consider the following features:

- Genre of the book including categories: fiction, drama, poetry, essay, collection.
- Number of pages in a book. The log transformation is applied to this parameter.
- Number of words in a book title. The log transformation is applied.
- Number of contributors (co-authors). The log transformation is applied.
- Using the author date of birth and publication year of the book, a new feature was engineered – the time interval between birth year and publication year of the book. The log transformation was applied.
- Number of fans for an author. This value was retrieved from Goodreads data. The log transformation is applied.

5 Prediction models

This is a regression task where the dependent parameter is average rating of a book. Mean squared error is used to compare the prediction model.

The genre category uses only one indicator: IsFiction. This change was made because Fiction category contains 2/3 of the dataset. The other categories are grouped together.

5.1 Regression models

5.1.1 Ordinary linear regression

The ordinary linear regression model resulted in MSE = 0.11. The residual plot is given in (Figure 10). The intercept and coefficients are:

Int.	Fict.	Pages	Title	Contr.	BP years	Fans	Ill.
3.47	-0.16	0.01	0.14	0.03	0.11	0.03	-0.03

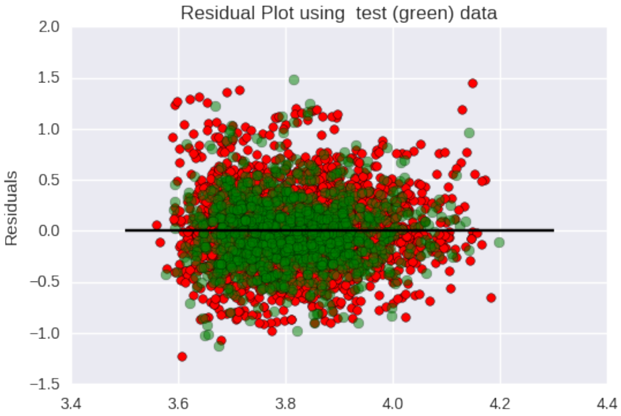

Figure 10: Residuals for ordinary linear regression model.

5.1.2 Ridge regression

Ridge regression model applies regularization coefficients to the decision function to prevent overfitting. Ridge regression model resulted in MSE = 0.11. The residual plot is given in (Figure 11). The intercept and coefficients are:

Int.	Fict.	Pages	Title	Contr.	BP years	Fans	Ill.
3.49	-0.16	0.01	0.13	0.03	0.10	0.03	-0.03

The coefficients do not differ much from the ones that resulted from ordinary regression.

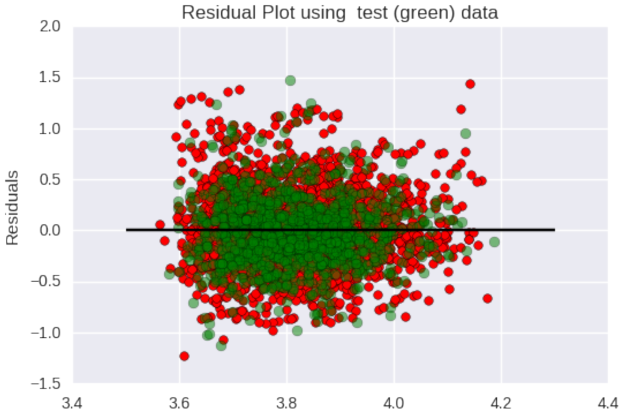

Figure 11: Residuals for ridge regression.

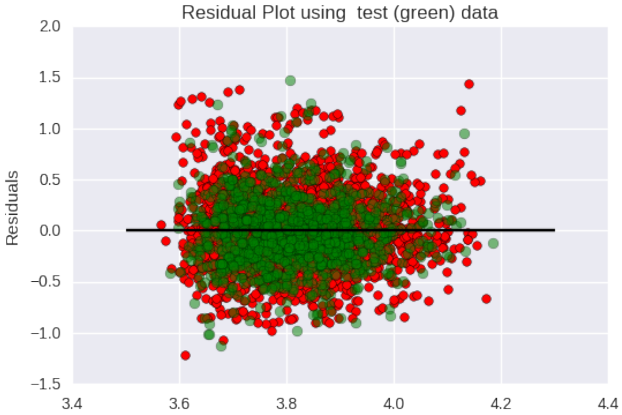

Figure 12: Residual plot for Bayes Ridge regression

5.1.3 Bayesian Ridge Regression

Bayesian techniques adapt to at hand. In ridge regression these techniques will adapt the regularization parameters based on the input data. Bayesian ridge regression yields MSE = 0.11. The residual plot is given in (Figure 12). The intercept and coefficients are:

Int.	Fict.	Pages	Title	Contr.	BP years	Fans	Ill.
3.5	-0.16	0.01	0.13	0.03	0.10	0.03	-0.03

5.1.4 Linear Support Vector Regression

The class of Support Vector Machines is successfully applied in classification tasks, but it can be extended to regression tasks as well. Linear Support

Vector regression yields MSE = 0.11. The residual plot is given in (Figure 13). The intercept and coefficients are:

Int.	Fict.	Pages	Title	Contr.	BP years	Fans	Ill.
3.42	-0.17	0.02	0.15	0.08	0.14	0.03	-0.02

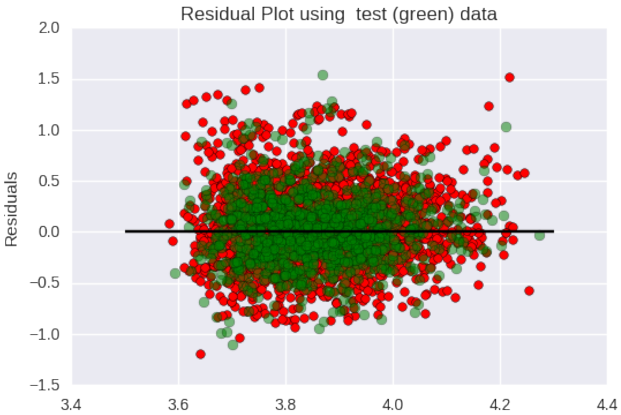

Figure 13: Residual plot for support vector regression

5.2 Regression trees

5.2.1 Decision Tree Regressor

Decision tree regressor performs poorly on the test data. The residual plot in (Figure 14) confirms the performance in training and testing records. MSE=0.24 is twice the value of other predictive models.

5.2.2 Random Forest Trees

Evaluated random forest trees with 300 estimators (trees). The model yielded MSE = 0.12. The residual plot in (Figure 15) depicts that the model performs well on the training data, but performs worse on the test data.

5.2.3 eXtreme Gradient Boosting (xgboost)

XGBOOST is one of the most popular ensemble methods used in many competitions.

In this work the result for XGBoostRegressor has the smallest MSE = 0.108. The residual plot in (Figure 16)

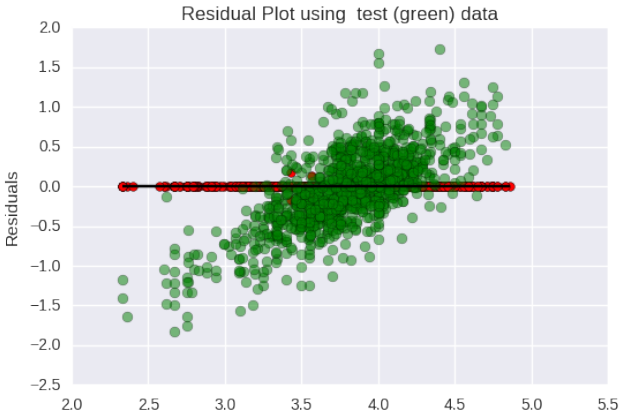

Figure 14: Residual plot for decision tree.

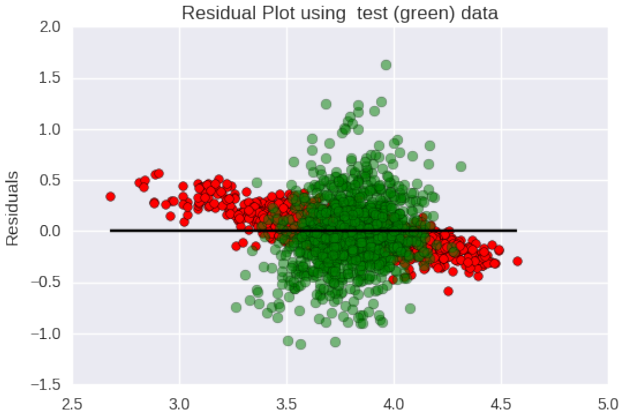

Figure 15: Residual plot for Random Forest Regressor

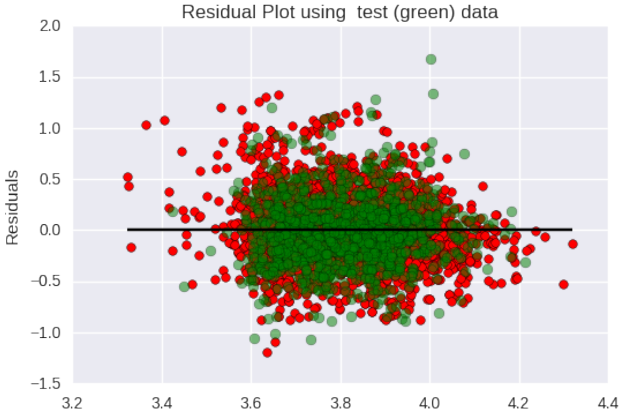

Figure 16: Residual plot for XGBoostRegressor

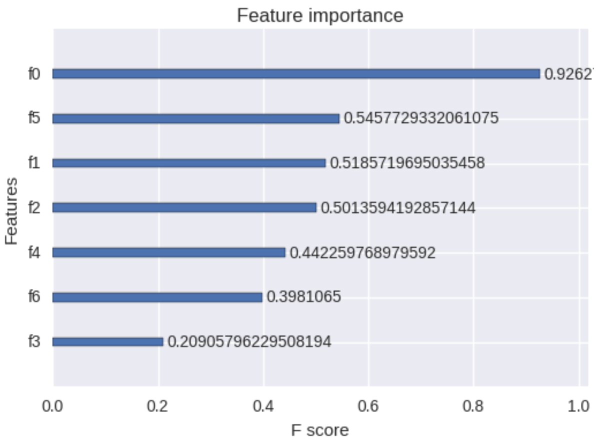

Figure 17: Feature importance plot from XGBoostRegressor

It is interesting to know the importance of the features. XGBoostRegressor provides this capability. The plot of important features is given in (Figure 17).

The results from importance plot bring some interesting insights. Three most important features in descending order are: genre, number of author fans, number of pages in a book. The number of words in the title might affect slightly the average rating of a book. Number of contributors and the fact that the book is illustrated are not very important features.

5.3 Hyper parameters performance tuning

The regressors above have fit the data with default values. We can tune the performance further by finding the model parameters that perform best. As established earlier, one of the best regressors is XGBoostRegressor. Will apply hyper-parameters tuning on this regressor.

```
max_depths = [2, 3, 4]
learning_rates = [0.04, 0.05, 0.07]
estims = [190, 200, 210]
gammas = [0, 0.01, 0.05]
colsamples_bytree= [0.85, 0.9, 0.95]
reg_lambdas = [9.0, 10.0, 15.0]

parameters = {'max_depth':max_depths, 'learning_rate':learning_rates, 'n_estimators':estims,
              'gamma':gammas, 'colsample_bytree':colsamples_bytree, 'reg_lambda':reg_lambdas}

clf = GridSearchCV(xgb, parameters, cv=3)
```

```

clf.fit(x_t_lr, y_lrn)

clf.best_estimator_
```

We use grid search on the regressor parameters. The values for each parameter are chosen over a wider range in the first iteration. We choose a cross validation set of 3 to evaluate the performance. The parameter values are refined in several passes. This procedure is time consuming. The best combination of the parameters is:

```
>>>XGBRegressor(base_score=0.5, colsample_bylevel=1, colsample_bytree=0.85,
       gamma=0.01, learning_rate=0.04, max_delta_step=0, max_depth=3,
       min_child_weight=1, missing=None, n_estimators=200, nthread=-1,
       objective='reg:linear', reg_alpha=0, reg_lambda=10.0,
       scale_pos_weight=1, seed=0, silent=True, subsample=1)
```

The tuned regressor performs better in terms of MSE. From 0.1083 it was improved to 0.1075. This is not a big improvement. The factor importance remained still the same.

6 Conclusions and recommendations

The purpose of this work is to get some insights on the parameters that make a book a bestseller. The criteria for a bestseller book has been associated with the readers' average rating score. If we are able to predict the average rating score for a book based on the available datasets, then a higher rating would indicate the success of a book.

We applied different predictive models and compared them based on MSE. The ordinary linear regression while simple is not performing worse than other advanced methods like Ridge Regression, Linear Support Vector regression or decision tree based models.

The standard error on test data for most of the models is 0.33. It means that we can make an error ±1 from the predicted value and be correct about it 99%. This imposes an issue when the prediction we make is around 3. The risk is that in reality a book with a rating of 3 could get an average rating of 2. As a result a bestseller would be a book predicted to have an average scoring rate closer to 4.

As per exploratory data analysis, we saw that a compilation of literature works yields a high rating. A collection of highly rated literature works has to be considered when publishing a new book. This is a relevant factor for publishers.

The datasets that we do have lack the information about the skills of an author or other literature metrics. Those would have been very interesting to work with. Retrieving this information from all the books is an interesting direction of research.

An important factor is also the audience the books are published for. The poetry is a highly rated genre. Fiction books are the most popular. Drama genre is slightly dominated by women in terms of higher ratings. Also women tend to write more on educational topics than men.

Genre and number of fans are important factors to take into consideration for a good rating.