R vs Python, why you should learn both?

Author: Suryansh Singh

Supervisor: Saikumar Allaka (senior data scientist at Quadratyx)

Introduction

This paper is aimed at those who are looking to break into the field of data science or have already done so, but are confused about which language they should learn first, R or python? This paper highlights the differences between the two and explains why it is beneficial to know both. On the internet, many people often offer their opinion about this subject from their personal experiences, however this differs from those as it presents proofs via case studies (done under the supervision of an experienced data scientist) as to which language is better and in what context. To determine the better language, both R and python were judged based on their performance and ease of usability with regards to topics like data analysis, time series analysis, natural language processing and machine learning.

Data Analysis

Data analytics is the process of inspecting, cleaning, transforming and modelling the data with the goal of discovering useful information to support decision making. While it includes many subfields, only a selected few have been covered in this paper.

Data preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format for future modelling and analytics purposes. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven process for resolving such issues and it prepares raw data for further processing.

For this task, the 'Lending club loan' dataset was used which contained noisy data. In order to preprocess the dataset, the less relevant variables were discarded using domain knowledge, missing values were imputed using central tendency measures of their respective variables, outlier values were identified and treated using a custom-made function, categorical variables containing more than 75% missing values were discarded (as they would not provide any useful insight for solving the problem), the remaining categorical variables were one-hot encoded (so that the computer could interpret them better) and finally to help the computer deal with imbalanced values, the whole dataset was normalized(scaled).

Language	Run-time (in seconds)
R	330.332
Python	43.257

Fig1.1

Language		Packages used
R		dummies, outliers, dplyr, DMwR
Python		sklearn

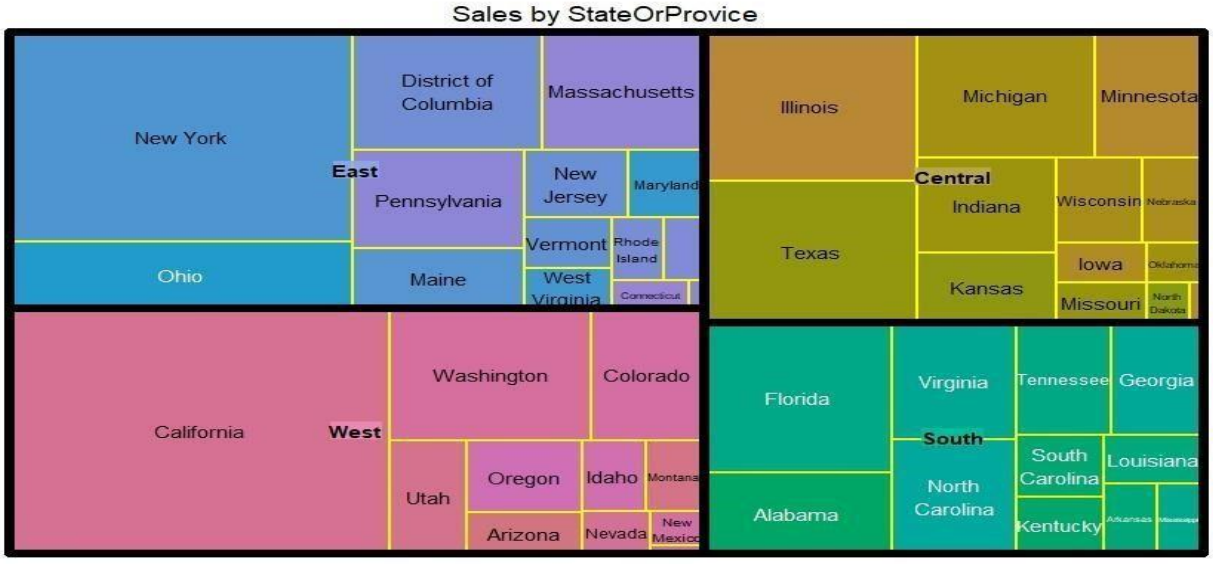
Fig1.2

By performing the process mentioned above, in both the languages, figures 1.1 and 1.2 were obtained. From those figures, it can be concluded that python is the language better suited for this task. Not only is python faster than R for carrying out data preprocessing tasks but the fact that it also contains all the preprocessing functionalities within a single external library gives it a massive advantage over R.

Exploratory data analysis

Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. Primarily, it is used for seeing what the data can tell us beyond the formal modelling or hypothesis testing task. EDA also encompasses initial data analysis, which includes checking assumptions required for model fitting and hypothesis testing and handling variable transformation.

For this task, the 'Sample-US Super Store' dataset form Tableau[1] was used and various graphs like bar plots, line charts, area charts, scatter plots, histograms, box and whisker plots, tree-maps, pie charts, bubble charts, word clouds, correlation plots and dual axis plots were plotted to explore and understand the data better.



(tree-map made in r)

Fig1.3

Language	Run-time (in seconds)
R	10.369
Python	29.228

Fig1.4

Language	Packages used
R	Plotly, treemap, wordcloud, ggplot2, corrplot
Python	Plotly, matplotlib, seaborn, wordcloud

Fig1.5

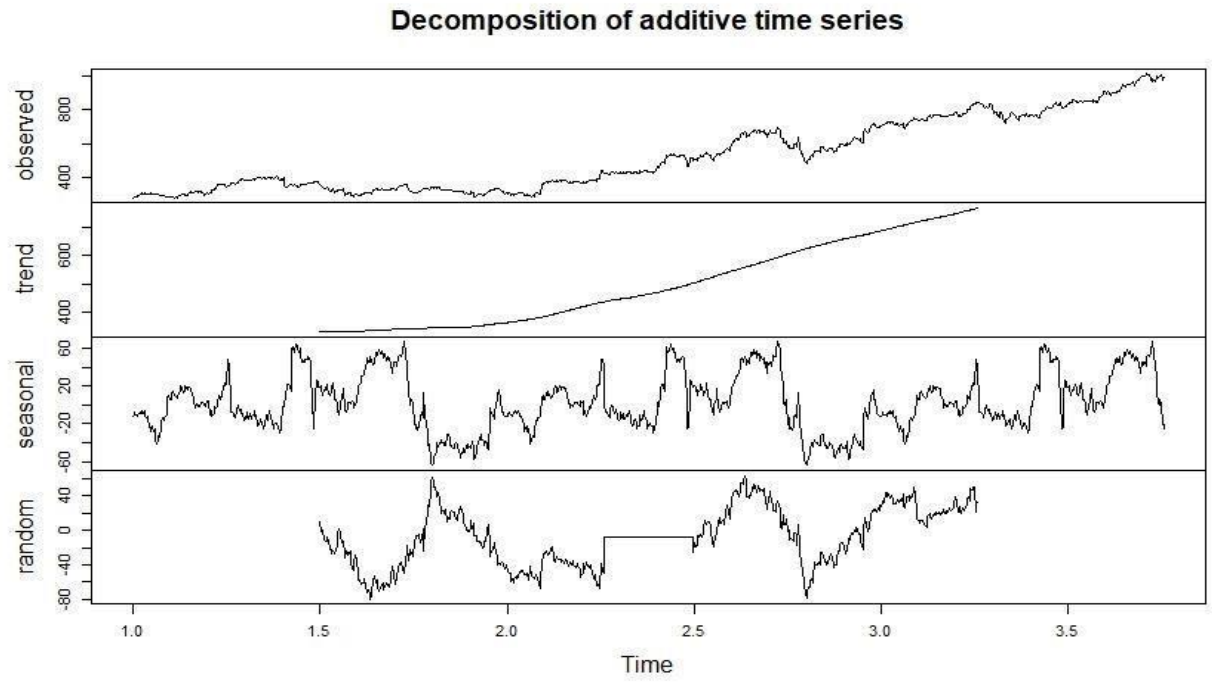
By making the various plots mentioned above, in both the languages, figures 1.3, 1.4 and 1.5 were obtained. Even though, R requires more number of packages to accomplish this task, it is quite easy to make various specific plots in it as compared to in python. For example, it requires more lines of code to make tree-maps and wordclouds in python, whereas they can be easily made in R with the help of simple single functions. Combining this observation with the results obtained in figure 1.4, it can be concluded that R is the more efficient language for this job, and since people don't want to spend too much time doing EDA, therefore R is the better suited language for this task.

Time-Series Analysis

A time series is a series of data points indexed in time order (E.g. stock data). Time series analysis comprises of methods for analyzing the time series data to extract meaningful statistics and other characteristics of the data. We can also predict future values in a time series by running predictive models on the analyzed data, this is known as time series forecasting.

For this task, a dataset containing the stock data for 20 different companies trading on NASDAQ was created using various data manipulation techniques. After the dataset was preprocessed, interactive time series charts were made to gain a basic understanding of the data. To analyze it even better, descriptive statistics for the data were calculated. Then decomposition techniques were used on the stock data to identify the pattern and seasonality as it would assist in time-series forecasting. For the ease of comparison, the dataset containing the stock data for Amazon was chosen for forecasting. This dataset was then transformed accordingly and holt-winters'[2] algorithm was implemented on it to predict the future closing stock prices.

Fig2.1



(made in R)

Fig2.2

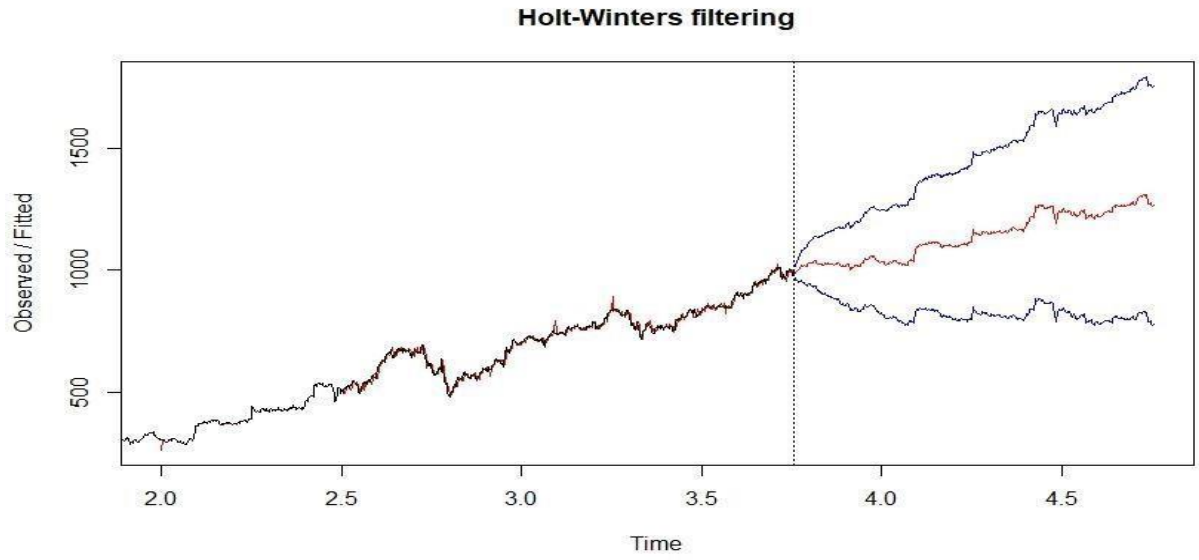


Fig2.3

Language	Run-time (in seconds)
R	4.733
Python	11.172

Fig2.4

Language	Packages used
R	plotly, psych, xts, dygraphs
Python	plotly, division, scipy, matplotlib, numpy

Fig2.5

By completing the process mentioned above, in both the languages, figures 2.1, 2.2, 2.3, 2.4 and 2.5 were obtained. From figures 2.4 and 2.5 it can be concluded that R is the language better suited for this task. Not only is it quicker to do time-series analysis in R but also comparatively easier. In R, complex forecasting algorithms can be implemented easily via simple functions, whereas python requires the user to write the code for implementing the various algorithms from scratch.

Natural language processing

Natural language processing (NLP) is a field of computer science, artificial intelligence and computational linguistics concerned with the interaction between computers and human languages, and, in particular, concerned with programming computers to fruitfully process large natural language corpora (samples).

For this task, a dataset containing the reviews for a restaurant taken from the internet was used. Before the dataset could be used for sentiment analysis, it was preprocessed using the necessary packages in both the languages. The preprocessing involved storage of all the reviews into a corpus, conversion of all the letters into lowercase (for uniformity) and the removal of numbers, punctuation, non-relevant words and white spaces from the corpus. After the preprocessing, a bag or words[3] model was created to facilitate the implementation of natural language processing on the reviews. For sentiment analysis of the reviews, the bag of words model was split into testing and training datasets. The appropriate classifiers were trained on the training set and then used to predict the sentiment of the reviews contained in the testing dataset. Finally, a confusion matrix was created to be used as a measurement of performance of the classifiers used.

Confusion matrix (in R)	0	1
0	79	21
1	30	70

(0 indicates a negative review while 1 indicates a positive review)

Fig3.1

Confusion matrix (in python)	0	1
0	87	10
1	46	57

(0 indicates a negative review while 1 indicates a positive review)

Fig3.2

Language	Run-time (in seconds)
R	4.052
Python	6.217

Fig3.3

Language	Packages used
R	tm, SnowballC, caTools, randomForest
Python	nltk, sklearn

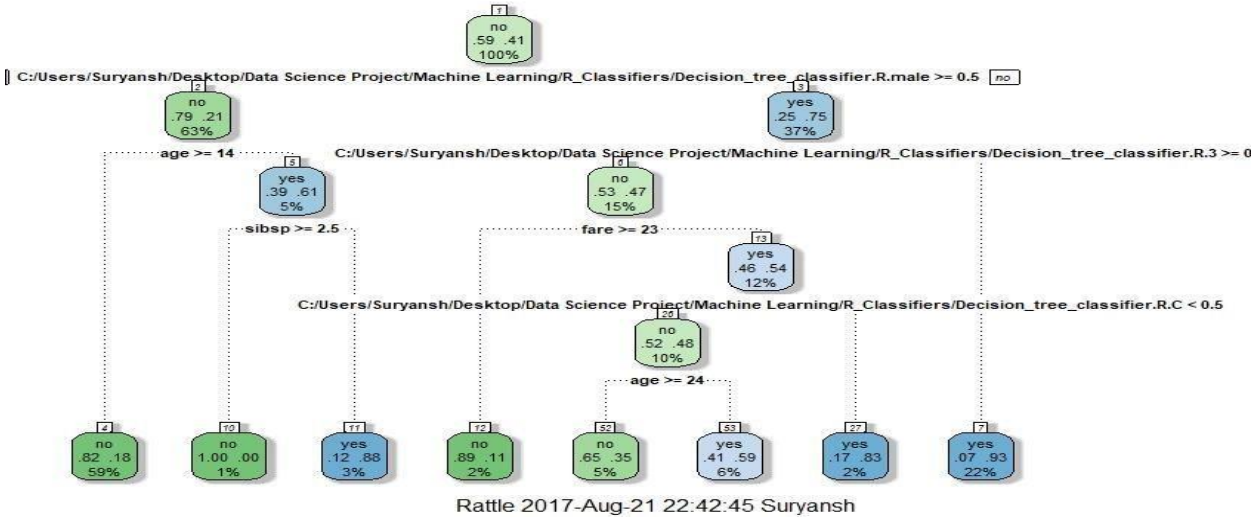
Fig3.4

By implementing the process mentioned above, in both the languages, figures 3.1, 3.2, 3.3 and 3.4 were obtained. From the results given in figures 3.3 and 3.4, it can be observed that while the runtime difference between the two languages isn't that significant, Python requires a considerably less number of packages to accomplish this task as compared to R. The NLTK package (in python) is also a very mature NLP package that enables python to handle many complex NLP tasks which may not be easily doable in R. By combining the results obtained from the figures given above with the fact that R does not have a similar package to NLTK, it can be concluded that Python is the better suited language for this task.

Machine Learning

Machine learning is the subfield of computer science that gives computers the ability to learn without being explicitly programmed. In this paper for the sake of argument, only classification algorithms like decision trees, k nearest neighbors, logistic regression, naïve bayes, support vector machines and random forests have been covered. In machine learning and statistics, classification is the problem of identifying which set of categories a new observation belongs to based on a training set of data containing observations whose category membership is known.

For this task, the 'Titanic: machine learning from disaster' dataset from Kaggle[4] was used. First the dataset was preprocessed before machine learning could be implemented on it. After preprocessing, the dataset was split into training and testing sets. The training set was used to train the different classifiers to enable them to make predictions on the data stored in the test dataset more accurately. To improve the performance of the models, k-fold validation was implemented on them, and by comparing the confusion matrices of the different models, the model with the highest accuracy was selected to make the predictions on the test set dataset.



(decision tree, for demonstration purposes)

Fig4.1

Classifier Name	Model accuracy before k-fold validation in python	Highest accuracy per fold in python	Average model accuracy in python after kfold validation*	Model accuracy before k-fold validation in R	Highest accuracy per fold in R	Average model accuracy in R after k-fold validation*
Logistic Regression	79.62%	89.16%	79.26%	77.40%	85.71%	79.17%
KNN	65.55%	67.86%	65.46%	75.00%	67.47%	64.19%
Decision Tree	70.81%	85.88%	79.12%	80.77%	83.13%	79.40%
SVM(linear kernel)	78.42%	84.34%	78.43%	74.52%	85.71%	78.31%
Naïve Bayes	77.82%	83.13%	77.22%	80.29%	85.71%	77.96%
Random Forests	79.42%	-	76.17%	80.29%	-	78.35%

*(Even though the average accuracy is going down it is not necessarily a bad thing as earlier the models may have been overfitting to the training set data.)

Fig4.2

Classifier Name	Run-time python (in seconds)	Run-time R (in seconds)
Decision tree	0.215	4.396
KNN	0.111	0.441
Logistic regression	0.202	0.917
Naïve Bayes	0.140	0.526
SVM	141.352	0.961
Random forest	0.106	0.775

(run-time includes the time taken for loading the dataset, preprocessing, model implementation and k-fold validation implementation)

Fig4.3

Language	Packages used
R	caret, e1071, randomforest
Python	sklearn

Fig4.4

By implementing the process mentioned above, in both the languages, figures 4.1, 4.2, 4.3 and 4.4 were obtained. From figures 4.2, 4.3 and 4.4 it can be concluded that python is the better suited language for this task. It is quicker to run machine learning models in python and the availability of various machine learning models and preprocessing functionalities within a single external library makes it a more efficient language to use.

Conclusion

From all the observations made above, the differences between the two languages is quite apparent. R excels at data/time-series analysis while python is better for tasks like machine learning, data preprocessing and natural language processing. The reason why R is so good at data analysis is because it is a statistical language made by statisticians. It already contains many inbuilt analytical functionalities for which python has to rely on external libraries. It also contains an astonishing number of external statistical packages which allow for very comprehensive data analysis unlike python. The quick nature of R code also makes it very useful for rapid prototyping in an industrial environment. However, it has its own limitations as well, like, sometimes it can be difficult to handle the large number of packages present in it or its slow run-time. On the other hand, since python is a generalpurpose programming language, it has a faster run-time and supports better product development. Due to a better memory management system, it also supports better machine learning, preprocessing and NLP functionalities and can handle larger datasets than R. Therefore, as a data scientist it is one's responsibility to pick the language that best fits their needs, thus, making it important for a data scientist to know about both the languages.

Future Work

In this paper, only a few selected topics like data preprocessing, exploratory data analysis, time-series analysis, natural language processing and machine learning have been covered to highlight the differences between the two languages. In the future, more advanced topics like deep learning, image processing, big-data processing, multi-processing, etc. can be covered for further comparing the differences between the two languages.

References

1. https://www.tableau.com/
2. https://www.otexts.org/fpp/7/5
3. https://en.wikipedia.org/wiki/Bag-of-words_model
4. https://www.kaggle.com/

Note: The relevant code used for the different sections of this paper can be found at https://github.com/Maverick2024/R-vs-Python-why-you-should-learn-both.git