MACHINE LEARNING WITH PYTHON

PREDICTING MOBILE APPS IN PLAYSTORE

TRAINING PROJECT REPORT

Submitted by:

**NAME – BHUMIKA JINDAL**

**UNIVERSITY ROLL NO. - 13000117112**

**B.TECH IN COMPUTER SCIENCE AND ENGINEERING**

**MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY,WEST BENGAL**



REPORT OF FOUR WEEKS INDUSTRIAL TRAINING

at

**WEBTEK LABS**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD

OF THE DEGREE OF **B.TECH**

(COMPUTER SCIENCE AND ENGINEERING)

CANDIDATE’S DECLARATION

**I hereby declare that I have undertaken industrial training at “WEBTEK LABS” during a period from 24 DECEMBER to 28 JANUARY in partial fulfilment of requirements for the award of degree of B.TECH in Computer Science and Engineering. The work which is being presented in the training report is an authentic record of training work.**

**Signature of the student**

**The three weeks industrial training Viva-voice examination of**

**Has been held on and accepted.**

**Signature of project mentor**

CERTIFICATE OF APPROVAL

The project “**PREDICTING MOBILE APPS IN PLAYSTORE**” made by **BHUMIKA JINDAL** is hereby approved as a creditable study for the Bachelor of Technology in Computer Science and Engineering and presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned this project only for the purpose for which it is submitted.

MS. MOUSITA DHAR

(Project In-charge)

ACKNOWLEDGEMENT

I would like to express our special thanks of gratitude to our trainer MS.MOUSITA DHAR who not only taught me about machine learning but also gave idea about its practical applications. She gave me the golden opportunity to do this wonderful project on the topic **PREDICTING MOBILE APPS IN PLAY STORE BY MACHINE LEARNING USING PYTHON** which also helped me in doing a lot of Research and to know about so many new things. I would like to thank her for her constant motivation and support to make me understand and complete my project.

I would also like to thank my college teachers who encouraged me to do the training. Finally, we would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

1. INTRODUCTION to python

**1.1 PYTHON**

Python is a high-level, interpreted, interactive and object-oriented scripting language. It is a high-level general-purpose, open source l, strictly typed programming language. The language provides constructs intended to enable clear programs on both a small and large scale. Python was created by Guido van Rossum. The Python Software Foundation (PSF) is the organization behind Python. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.
* **PYTHON FEATURES**
* **Easy** to Learn and Use. Python is **easy** to learn and use
* Expressive Language
* Dynamic Language
* Interpreted Language
* Cross-**platform** Language
* Free and Open Sourced
* Multipurpose
* Strongly typed
* **Object-Oriented** Language
* Extensible
* Large Standard Library
* **APPLICATIONS OF PYTHON**
* Web development
* Internet Of Things(IOT)
* Scientific and numeric computing
* Data Analysis
* Desktop GUIs
* Machine Learning
* Image Processing
* Data visualization
* Game Development
* Software Development
* Business Application
  1. **ANACONDA**

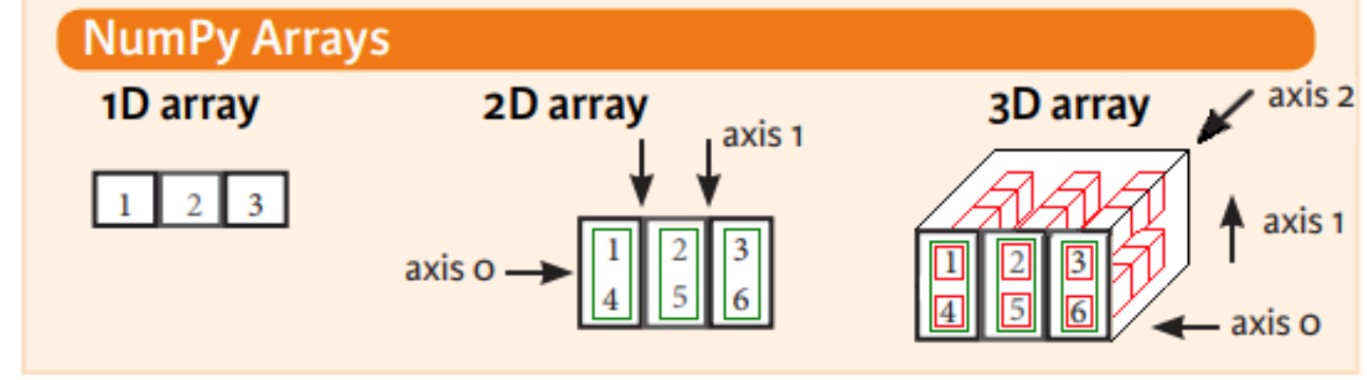
Anaconda is a free and open distribution of Python programming languages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. It includes hundreds of popular data science packages and the *conda* package and virtual environment manager for Windows, Linux, and MacOS. Package versions are managed by the package management system *conda*. Conda is an open source, cross platform, language-agnostic package manager and environment management system that installs, runs and upgrade complex data science and machine learning environments like scikit-learn, TensorFlow, and SciPy. Anaconda Distribution is the foundation of millions of data science projects as well as Amazon Web Services. The Anaconda distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and MacOS.

* 1. **IPYTHON**

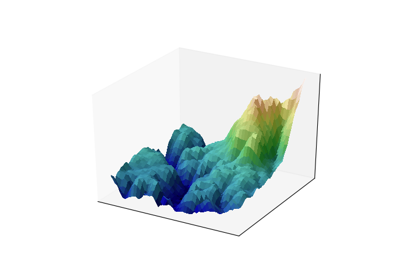
Python (Interactive Python) is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers introspection, rich media, shell syntax, tab completion, and history. IPython provides the following features:

* Interactive shells (terminal and Qt-based).
* A browser-based notebook interface with support for code, text, mathematical expressions, inline plots and other media.
* Support for interactive data visualization and use of GUI toolkits.
* Flexible, embeddable interpreters to load into one's own projects.
* Tools for parallel computing.
  1. **PYTHON PACKAGES**
* **NumPy**
* NumPy is the fundamental package for scientific computing with Python.
* a powerful N-dimensional array object
* sophisticated (broadcasting) functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

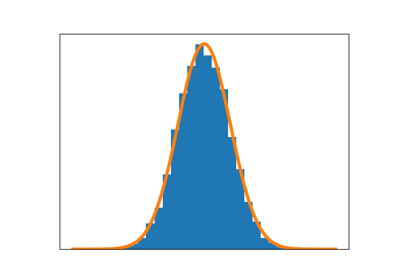


* **Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.



* **Pandas**

*Pandas* is an open source, BSD-licensed library providing high-performance, easy- to-use data structures and data analysis tools for the *Python* programming language. Pandas library is well suited for data manipulation and analysis using python. In particular, it offers data structures and operations for manipulating numerical tables and time series.



* **Scikit-learn**

Scikit-learn provides machine learning libraries for python some of the features of Scikit- learn includes:

* Simple and efficient tools for data mining and data analysis
* Accessible to everybody, and reusable in various contexts
* Built on NumPy, SciPy, and matplotlib
* Open source, commercially usable - BSD license



* **Seaborn**

Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

1. TRAINING WORK UNDERTAKEN

**INTRODUCTION**

There are many services today which provide users with information about a large number of different items. Good examples of this are systems used in electronic commerce, various public databases, and many others. Information about items that might appear more interesting to users is used to improve the quality of such services. The user ratings given to some items are used to determine the ratings of other items. Those systems which perform such tasks are called automatic ratings prediction systems. Automatic ratings prediction systems determine a possible user rating for a specific item using a set of chosen features. The information about a possible user rating can be used for different purposes. For example, bookselling services can identify books more interesting to users and use this information to increase sales. Similarly, while using public databases, a user can be presented with data that might appear more interesting to him, thus improving service quality. Good ratings prediction requires the correct interpretation of the available data about both the user and the item. Therefore, the design of such systems often uses information retrieval and machine learning techniques to find regularities in the available data. The techniques used can differ in their complexity, the amount of data required, as well as other features.

This paper presents a short overview of some automatic ratings prediction methods. The chosen methods are used in automatic ratings prediction systems of many well-known services and are applied to various items. The paper is based on Mobile apps rating prediction, so an adequate dataset was collected, the chosen methods were implemented and evaluated, and the acquired results were analysed. The paper is organised as follows. The next section presents a short overview of related papers and research on the subject. Then, the descriptions of the used ratings prediction methods are given, divided by the features they use. This is followed by the description of the evaluation process, as well as the presentation and analysis of the results. The last section concludes the paper.

Ratings prediction is closely related to the problem of recommending items to users. A lot of research into this problem has already been done. However, most of this research reduces the problem of recommendation to the problem of predicting a suitably defined rating for the user and then recommends the item with the highest rating. Three different approaches have emerged in the research of ratings prediction: content-based methods, collaborative methods, and hybrid methods. Content-based methods predict user ratings using the item’s features and the user’s affinities, while collaborative methods predict ratings using the ratings of other, similar users. Hybrid methods combine the two aforementioned approaches. These approaches can additionally be subdivided into heuristic and model based approaches.

The App Store is a digital distribution platform, developed and maintained by Apple Inc., for mobile apps on its iOS operating system. The store allows users to browse and download apps developed with Apple's iOS software development kit. Apps can be downloaded on the iPhone smartphone, the iPod Touch handheld computer, or the iPad tablet computer, and some can be transferred to the Apple Watch smartwatch or 4th-generation or newer Apple TVs as extensions of iPhone apps.The App Store was opened on July 10, 2008, with an initial 500 applications available. As of 2017, the store features over 2.1 million apps. Developers have multiple options for monetizing their applications, ranging from free, free with in-app purchases, and paid. However, App Store has been criticized for a lacklustre development environment, prompting the company in June 2016 to announce a "renewed focus and energy" on the store. Major changes introduced in the following months include ads in search results, a new app subscription model, and the ability for developers to respond to customer reviews. Additionally, Apple began a process to remove old apps that do not function as intended or that don't follow current app guidelines, with app research firms noticing significant numbers of app removals from the store. Furthermore, with the release of iOS 11 in September 2017, App Store received a complete design overhaul, bringing a greater focus on editorial content and daily highlights, as well as a design similar in style to several of Apple's built-in iOS apps.

**PROBLEM DESCRIPTION**

**Given a set of user review data each of which is associated with a user rating, the objective is to build a model that is able to predict numerical rating from textual data having various factors on which the total rating of the app depends which includes app size, downloads, genre, rating, versions, price, number of languages supported and screenshots available showing various features.**

**Inspiration/Motivation to do this project:**

* How does the App details contribute the user ratings?
* Try to compare app statistics for different groups?

**Reference: R package** From github, with devtools::install\_github("ramamet/applestoreR")

The data was extracted from the [iTunes Search API](http://www.transtats.bhttps/developer.apple.com/library/content/documentation/AudioVideo/Conceptual/iTuneSearchAPI/SearchExamples.html#//apple_ref/doc/uid/TP40017632-CH6-SW1ts.gov/DatabaseInfo.asp?DB_ID=120&Link=0) at the Apple Inc website. R and linux web scraping tools were used for this study.

I have tried various machine learning algorithms on the given dataset which includes linear regression, decision tree and random forest and accepting the algorithm which yields highest accuracy and gives the best results.

# **Mobile App Statistics (Apple iOS app store)**

The ever-changing mobile landscape is a challenging space to navigate. The percentage of mobile over desktop is only increasing. Android holds about 53.2% of the smartphone market, while iOS is 43%. To get more people to download your app, you need to make sure they can easily find your app. Mobile app analytics is a great way to understand the existing strategy to drive growth and retention of future user. With millions of apps around nowadays, the following data set has become very key to getting top trending apps in iOS app store. This data set contains more than 7000 Apple iOS mobile application details. The data was extracted from the [iTunes Search API](http://www.transtats.bhttps/developer.apple.com/library/content/documentation/AudioVideo/Conceptual/iTuneSearchAPI/SearchExamples.html#//apple_ref/doc/uid/TP40017632-CH6-SW1ts.gov/DatabaseInfo.asp?DB_ID=120&Link=0) at the Apple Inc website. R and linux web scraping tools were used for this study.

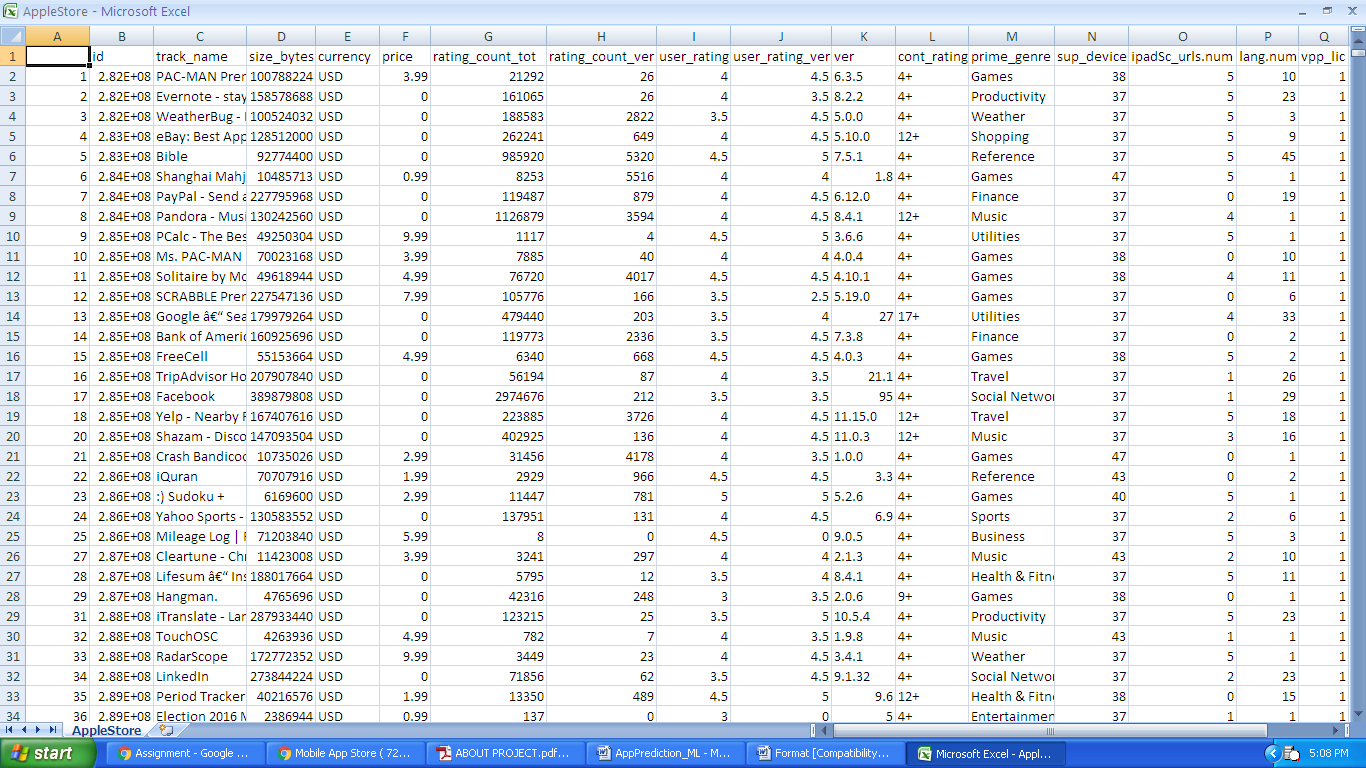
**Data collection date (from API)**; July 2017

**Dimension of the data set**: 7197 rows and 16 columns

## ****Content:****

## appleStore.csv

1. "**id**" : App ID
2. "**track\_name**": App Name
3. "**size\_bytes**": Size (in Bytes)
4. "**currency**": Currency Type
5. "**price**": Price amount
6. "**rating\_count\_tot**": User Rating counts (for all version)
7. "**rating\_count\_ver**": User Rating counts (for current version)
8. "**user\_rating**" : Average User Rating value (for all version)
9. "**user\_rating\_ver**": Average User Rating value (for current version)
10. "**ver**" : Latest version code
11. "**cont\_rating**": Content Rating
12. "**prime\_genre**": Primary Genre
13. "**sup\_devices.num**": Number of supporting devices
14. "**ipadSc\_urls.num**": Number of screenshots showed for display
15. "**lang.num**": Number of supported languages
16. "**vpp\_lic**": Vpp Device Based Licensing Enabled

****

**THE LEARNING SYSTEM**

* **COLLECTING DATA FROM KAGGLE**

**Kaggle** is a platform for predictive modelling and analytics competitions in which statisticians and data miners compete to produce the best models for predicting and describing the datasets uploaded by companies and users. This crowd sourcing approach relies on the fact that there are countless strategies that can be applied to any predictive modelling task and it is impossible to know beforehand which technique or analyst will be

most effective.

On 8 March 2017, Google announced that they were acquiring Kaggle. They will join the Google Cloud team and continue to be a distinct brand. In January 2018, Booz Allen and Kaggle launched Data Science Bowl, a machine learning competition to analyze cell images and identify nuclei.

* **DATA SCIENCE**

**Data science** is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, structured and unstructured, similar to data mining. Data science is a "concept to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyze actual phenomena" with data. It employs techniques and theories drawn from many fields within the context of mathematics, statistics, information science, and computer science.

Turing award winner Jim Gray imagined data science as a "fourth paradigm" of science (empirical, theoretical, computational and now data-driven) and asserted that "everything about science is changing because of the impact of information technology" and the data deluge.

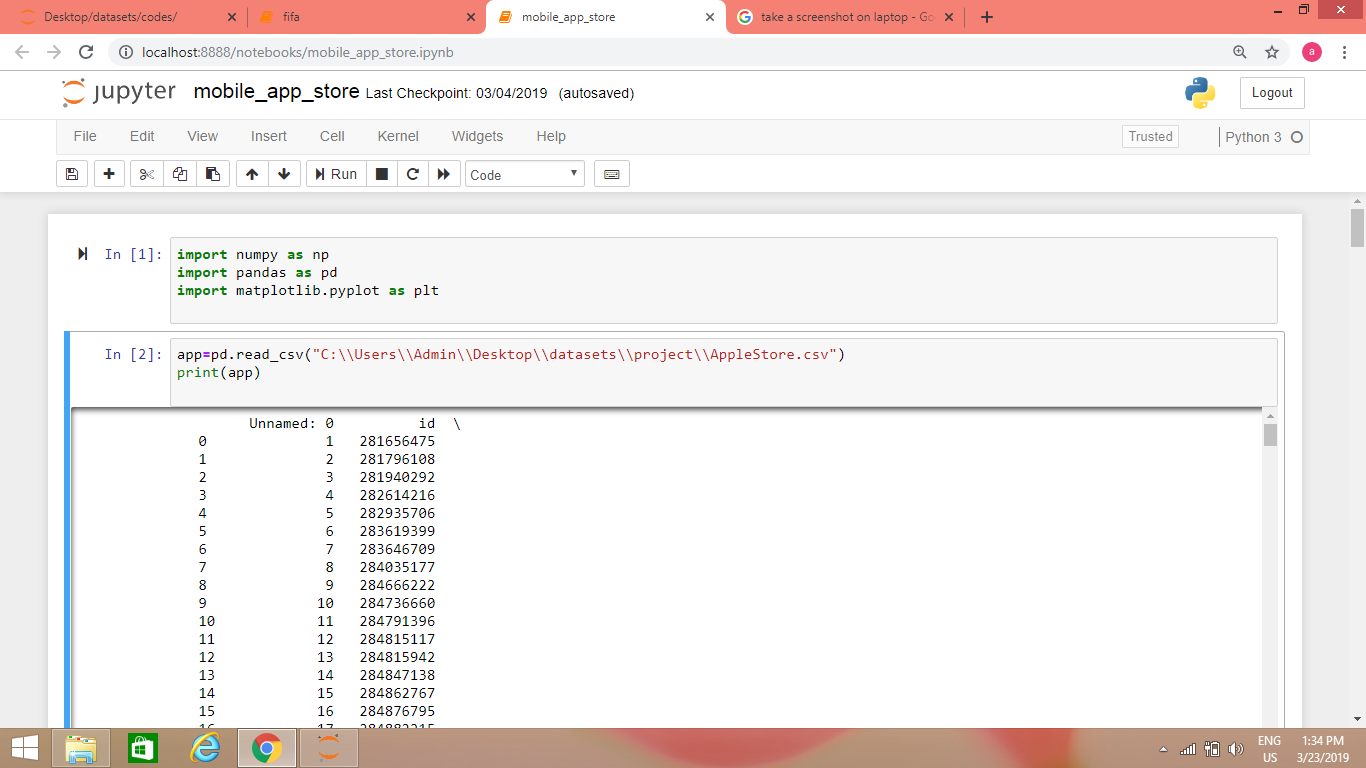
When Harvard Business Review called it "The Sexiest Job of the 21st Century" the term became a buzzword, and is now often applied to business analytics, business intelligence, predictive modelling, or any arbitrary use of data, or used as a glamorized term for statistics. In many cases, earlier approaches and solutions are now simply rebranded as "Data science" to be more attractive, which can cause the term to become "dilute[d] beyond usefulness." While many university programs now offer a data science degree, there exists no consensus on a definition or suitable curriculum contents. Because of the current popularity of this term, there are many "advocacy efforts" surrounding the field. To its discredit, however, many data science and big data projects fail to deliver useful results, often as a result of poor management and utilization of resources.

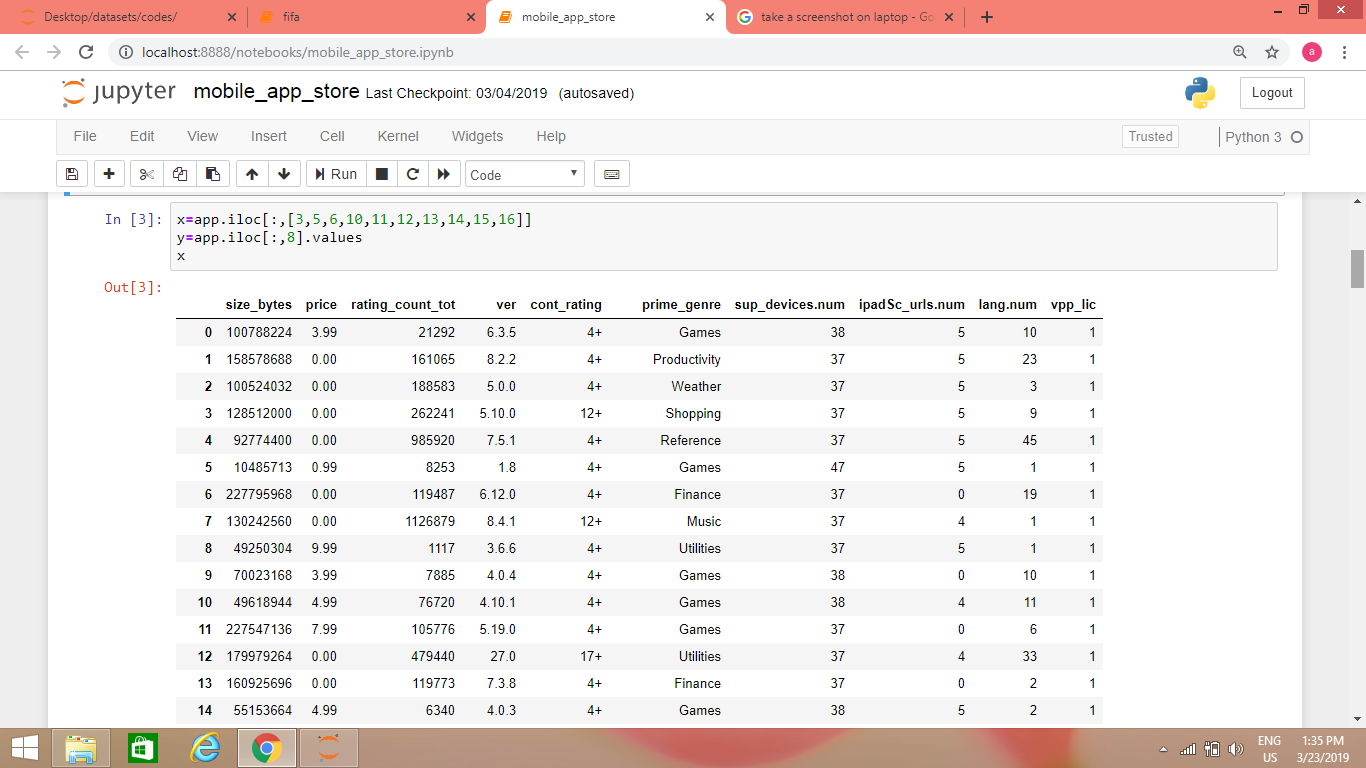
**PREDICTING MOBILE APPS IN PLAY STORE**

In this project, I intend to use mobile data of several iOS apps to predict the average user ratings for different apps and find out which apps are the best and worst performers. From a **technical point of view**, the main aspects of python used throughout the project are:

* **Visualization:** Matplolib
* **Data manipulation:** pandas, numpy
* **Modeling:** Sklearn
* **Class Definition:** Random Forest, Decision Tree, Logistic Regression

**SOURCE CODE & OUTPUT:**

* **Importing numpy, pandas and matplotlib modules**
* **Load the data i.e. read the given .csv file of dataset**
* **Splitting the columns of the given dataset to find x and y**

Among 16 columns present in the dataset, user\_rating which is in 8th column needs to be predicted, so it is stored in a predictor variable, y. Other factors on which the ratings will depend is stored in independent variable named x.

* **Data pre-processing:**
* **Missing value imputation**

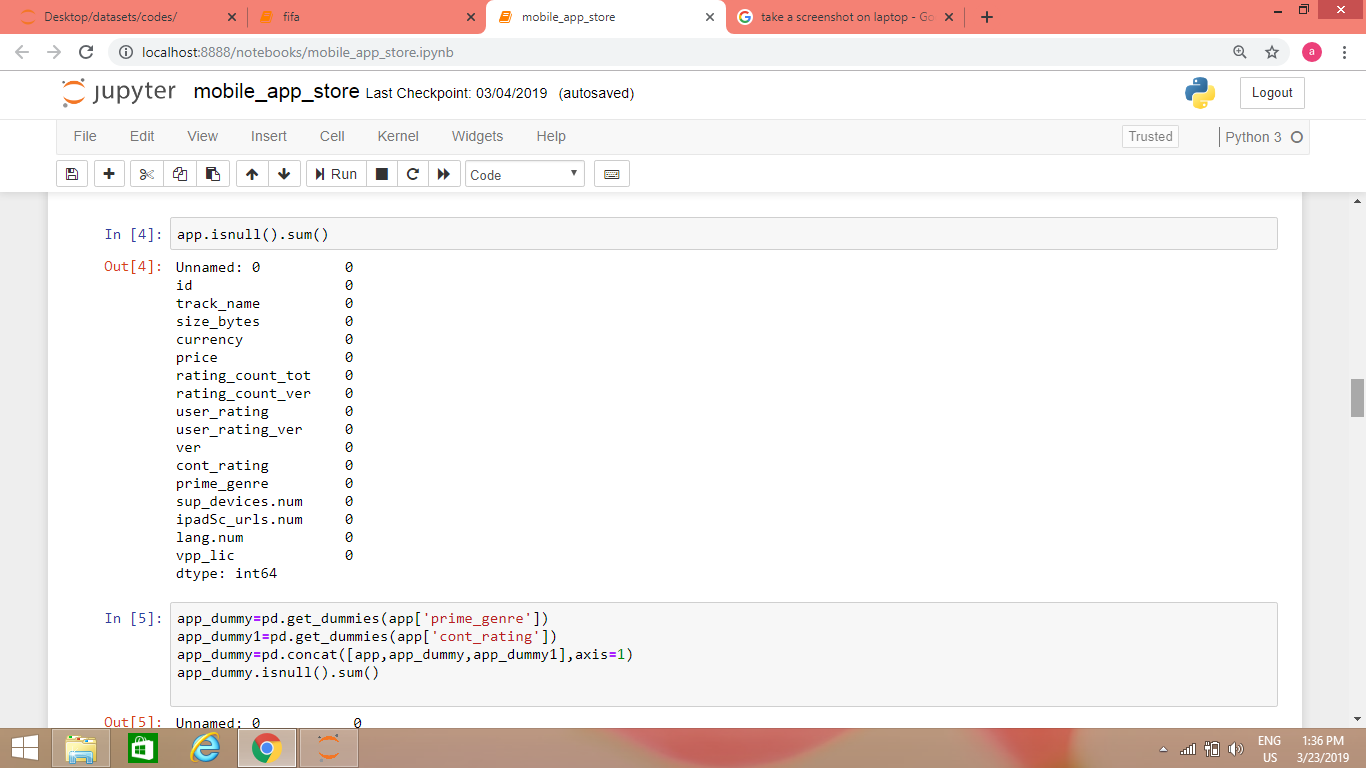
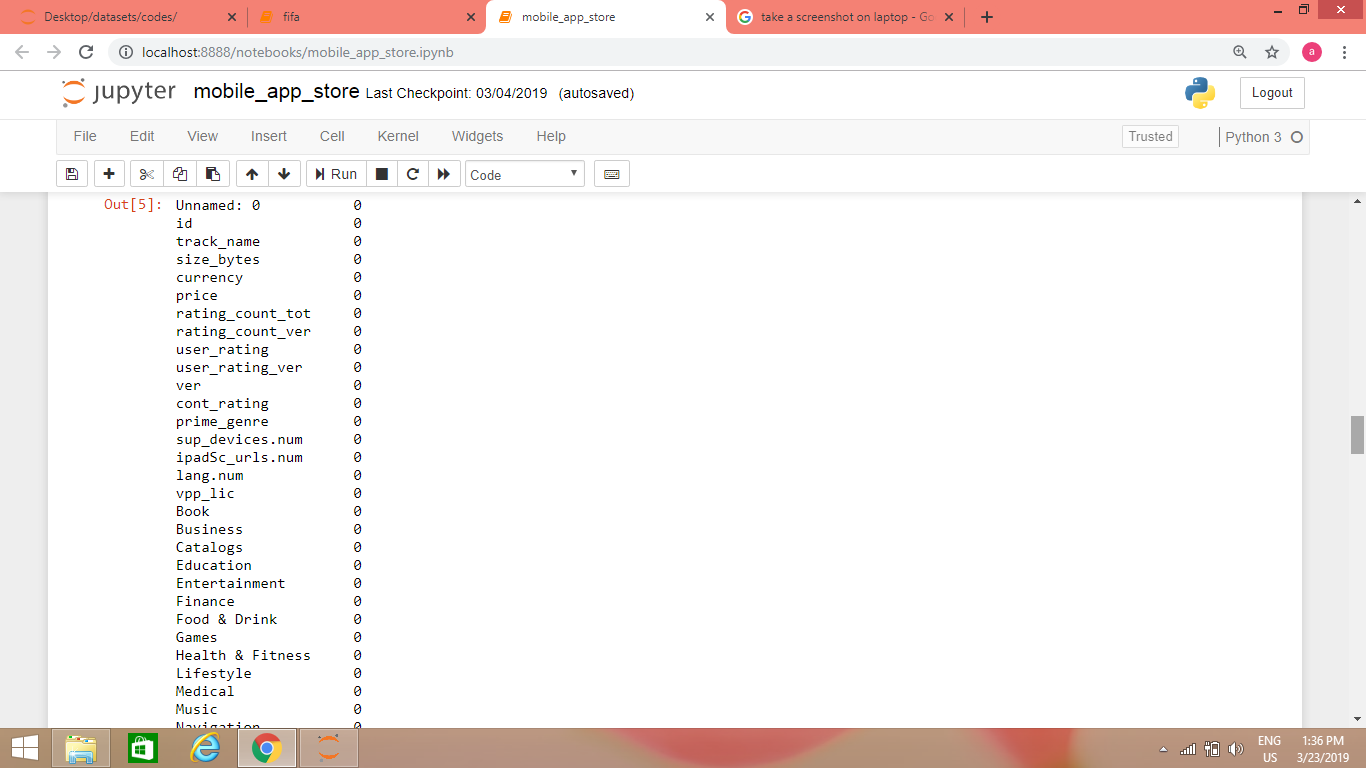
After checking for null values, it is seen that the data set doesn’t contain any null values so there are no missing values present, hence no need to impute.

* **Encoding categorical variables**

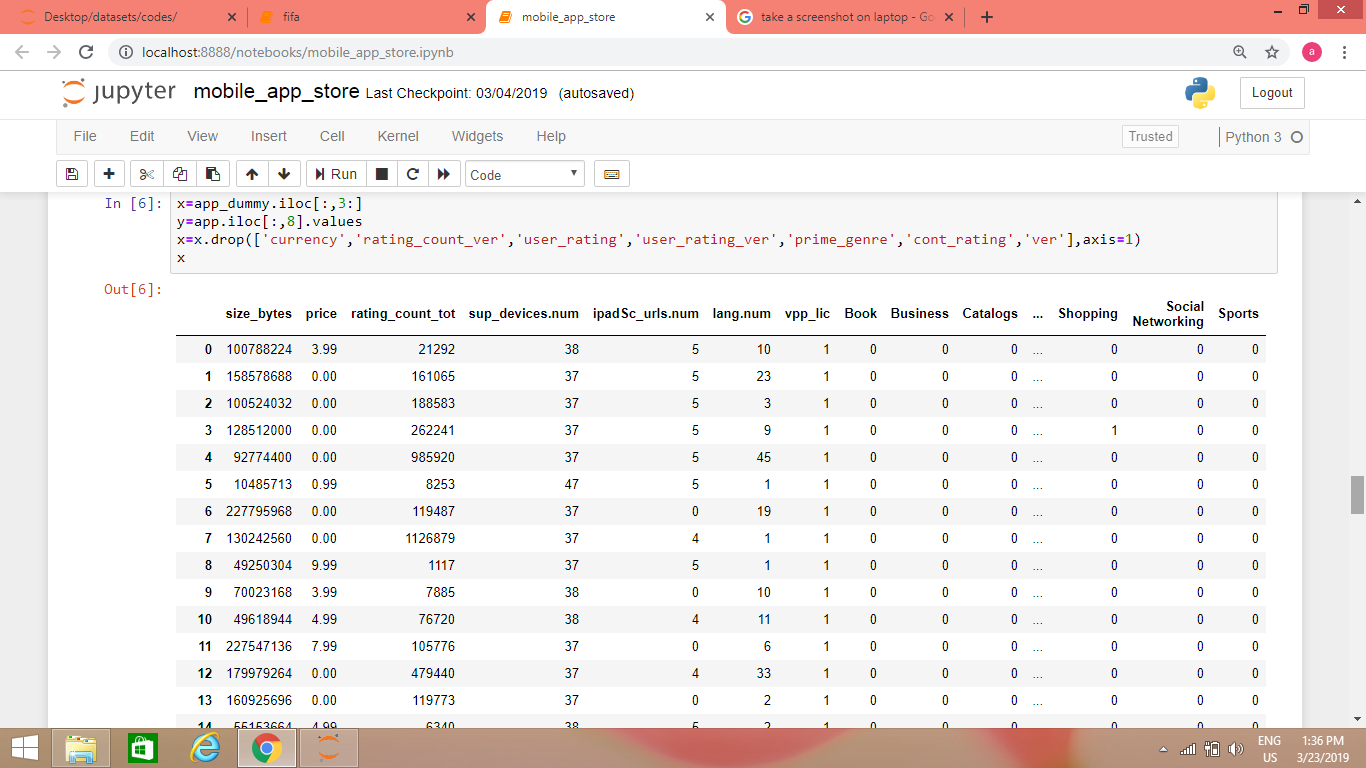
After checking for String values, it is seen that the data set contain string values other than numeric values in prime\_genre and cont\_rating so they are converted to numeric types by using get\_dummies. Thus the new columns formed by them are concatenated to the original dataset.

* **Scaling using Standard Scaler**

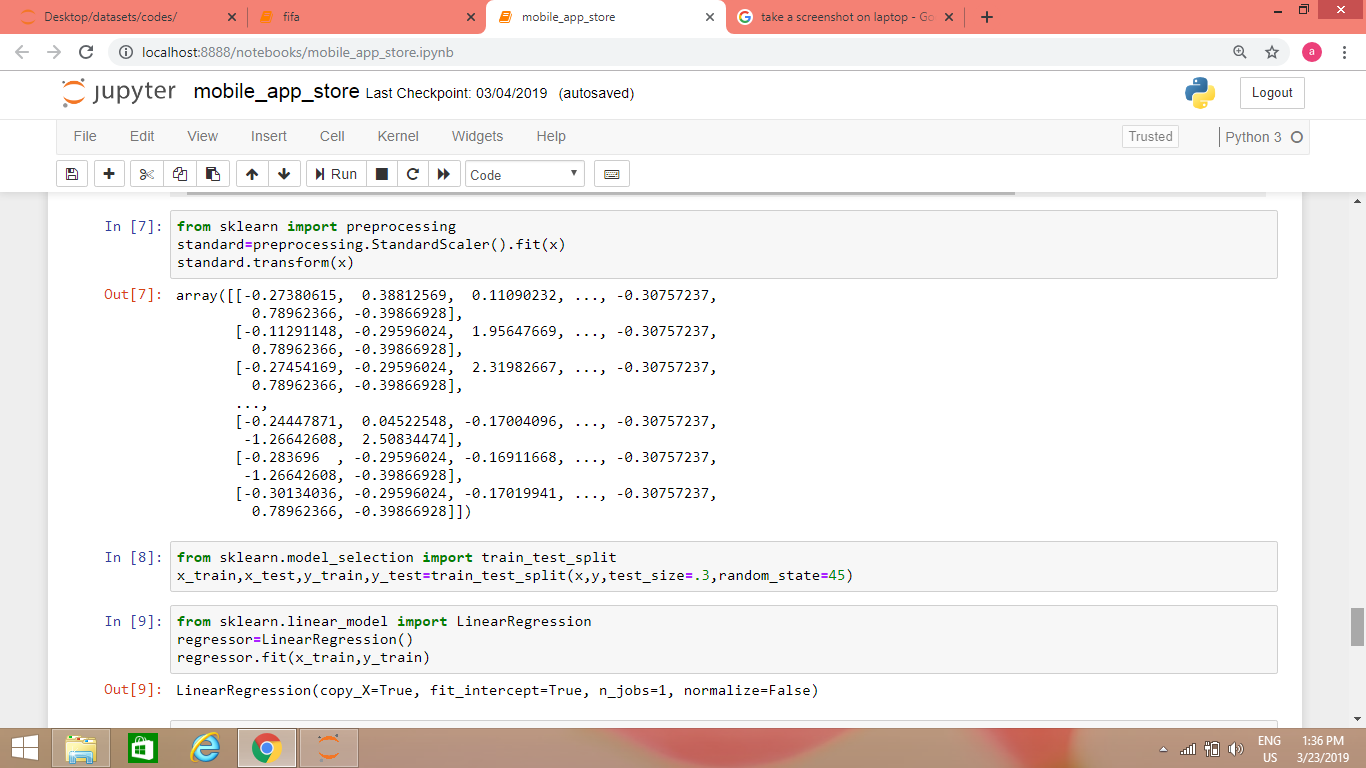
After checking for the variation in different rows of a particular column, it is seen that there is variation in some columns. So the data is scaled using Standard Scaler.

****

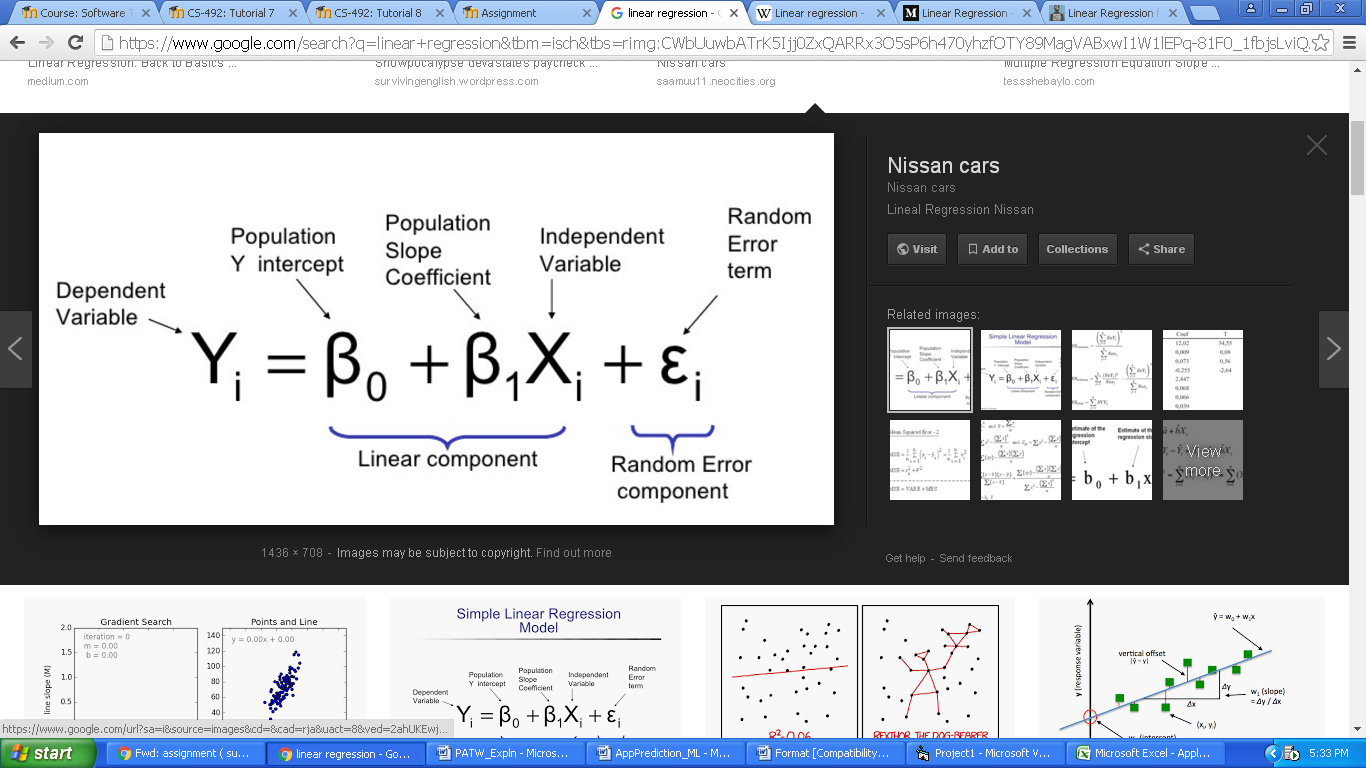
* **Splitting the columns of the modified dataset to find x and y**

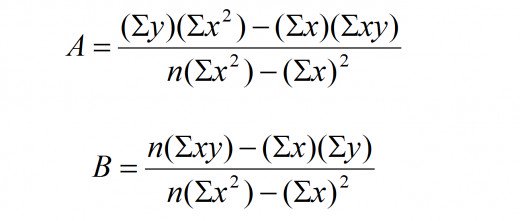
Among the columns present in the modified dataset, user\_rating which is in 8th column needs to be predicted, so it is stored in a predictor variable, y. Other factors on which the ratings will depend is stored in independent variable named x. Also, the data is cleaned by removing or dropping some of the columns of factors having negligible role or which will affect the model.

* **Splitting the dataset into training and testing dataset**

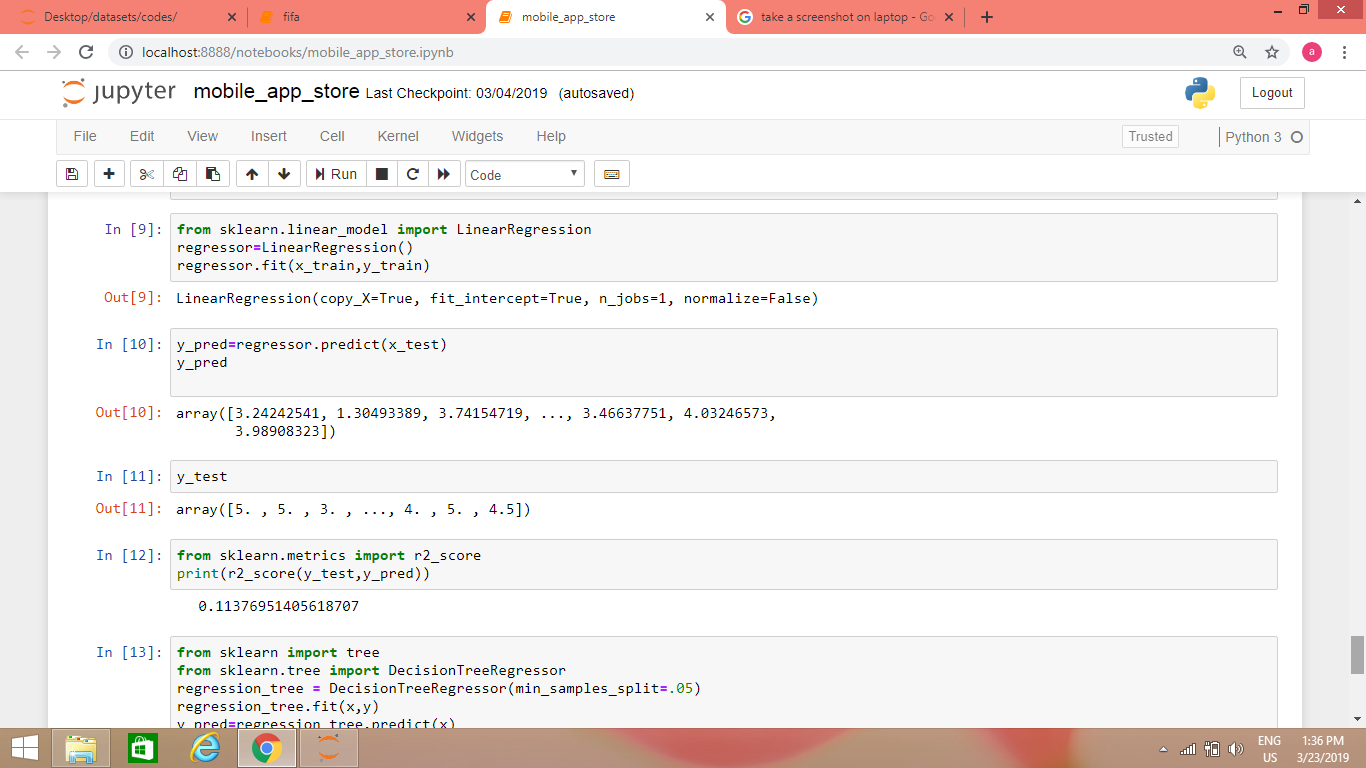
After scaling the data, the dataset is divided into two parts- training dataset which is used to make our model and testing dataset which is used to test the model. This is done by train\_test\_split function of sklearn.

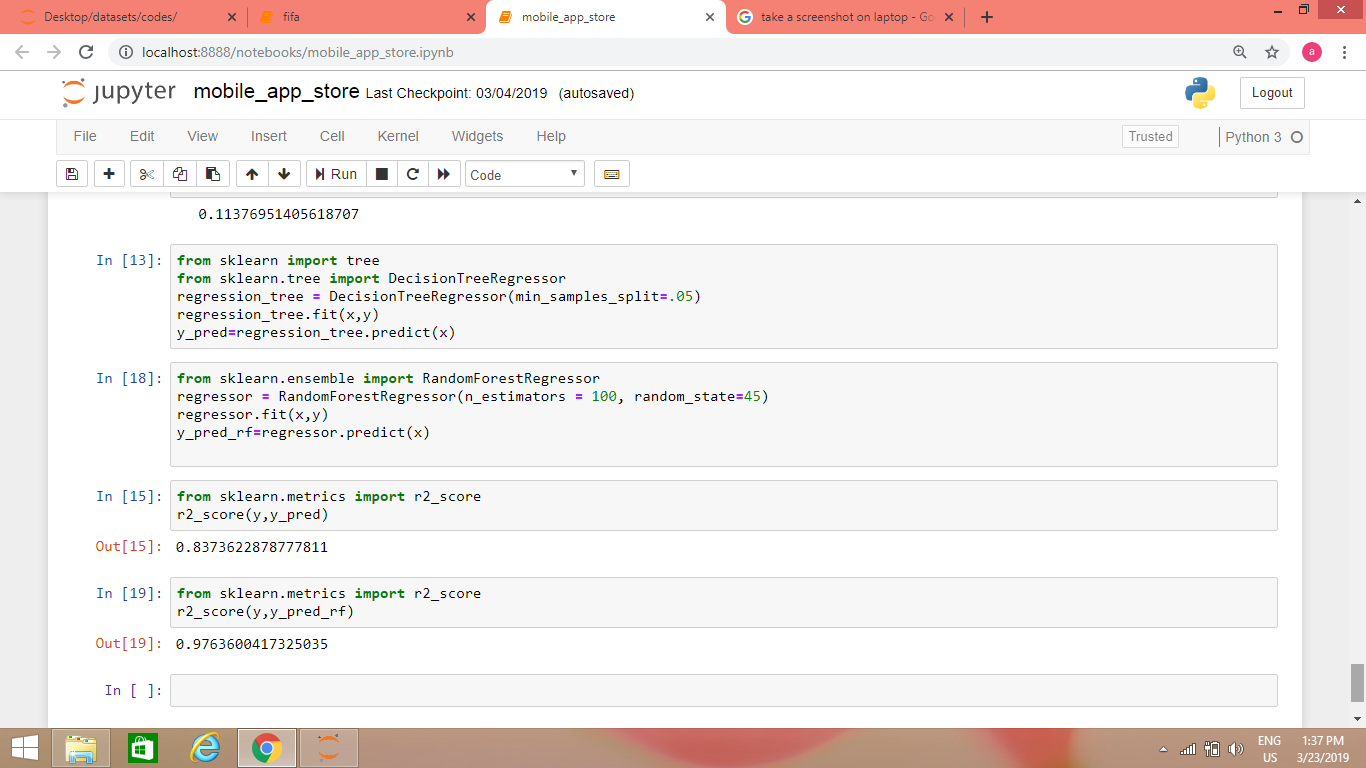
**DIFFERENT LEARNING ALGORITHMS USED AND THEIR RESULTS**

* **LINEAR REGRESSION**
* Linear Regression is used as a model for understanding the relationship between input and output numerical variables, but has been borrowed by machine learning. It is both a statistical algorithm and a machine learning algorithm.
* A regression problem requires the prediction of a quality.
* Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variable(x) and the single output variable(y).
* When there is only one x and one y then it is called simple linear regression and when y depends on more than one independent variable(x) then it is multiple linear regression. Here, we will be using multiple linear regression.
* SKLearn has a method LinearRegression() using which we can fit regression model and predict the data.It will calculate both coefficient and intercept. Then we fit the model with x and y.
* This is the formula for linear regression, where A is the slope and B is the intercept.



* ***This is the input and output code for Linear Regression Algorithm. Firstly, we import the LinearRegression module from SKLearn and then fitting the training dataset to the model. Then the values of y is predicted on testing dataset. The accuracy score is found using r2\_score and it is 11.37% only. This shows that this algorithm is not a good one to be used.***

****

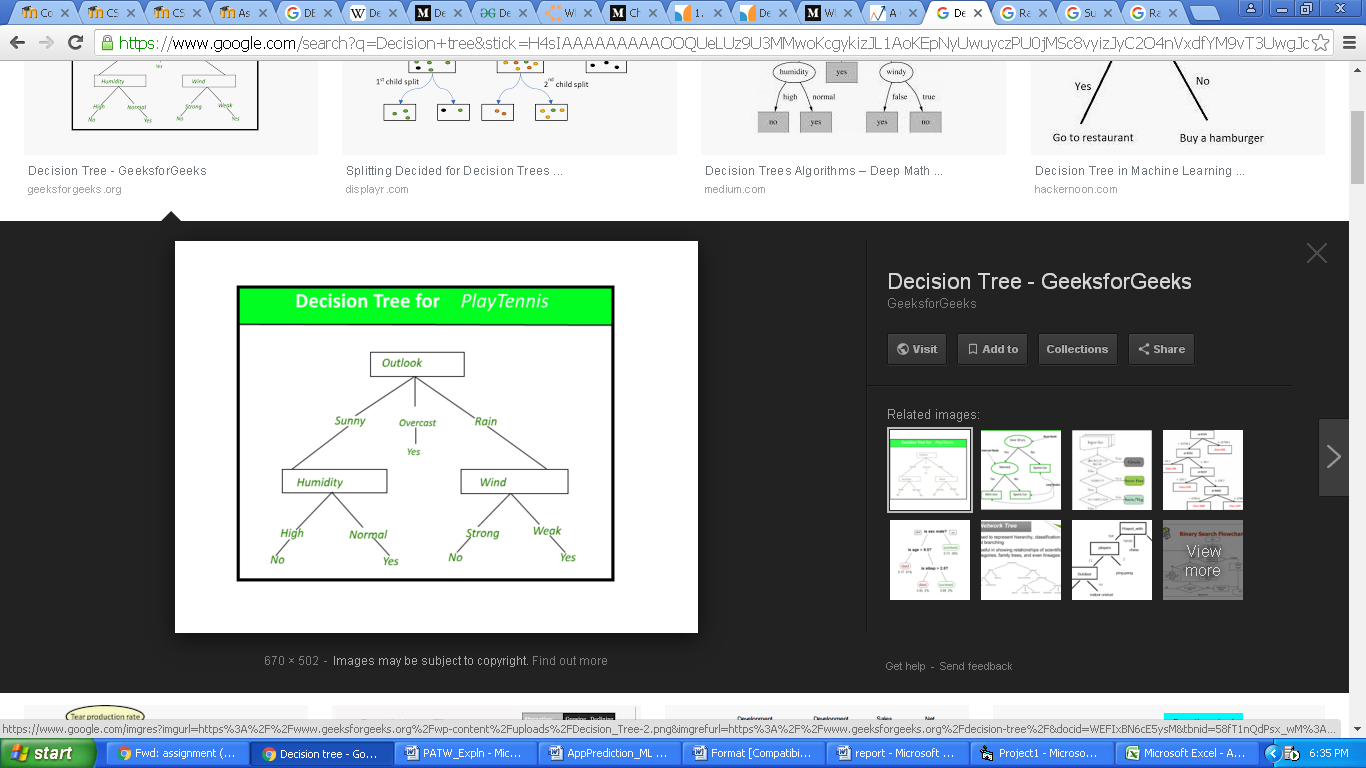
****

* **DECISION TREE CLASSIFIER**
* Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.
* **Construction of Decision Tree:**

A tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

* **Decision Tree Representation :**

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the subtree rooted at the new node.

The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf.(in this case Yes or No). For example, the instance:

(Outlook = Rain, Temperature = Hot, Humidity = High, Wind = Strong )

would be sorted down the leftmost branch of this decision tree and would therefore be classified as a negative instance.

In other words we can say that decision tree represent a disjunction of conjunctions of constraints on the attribute values of instances.

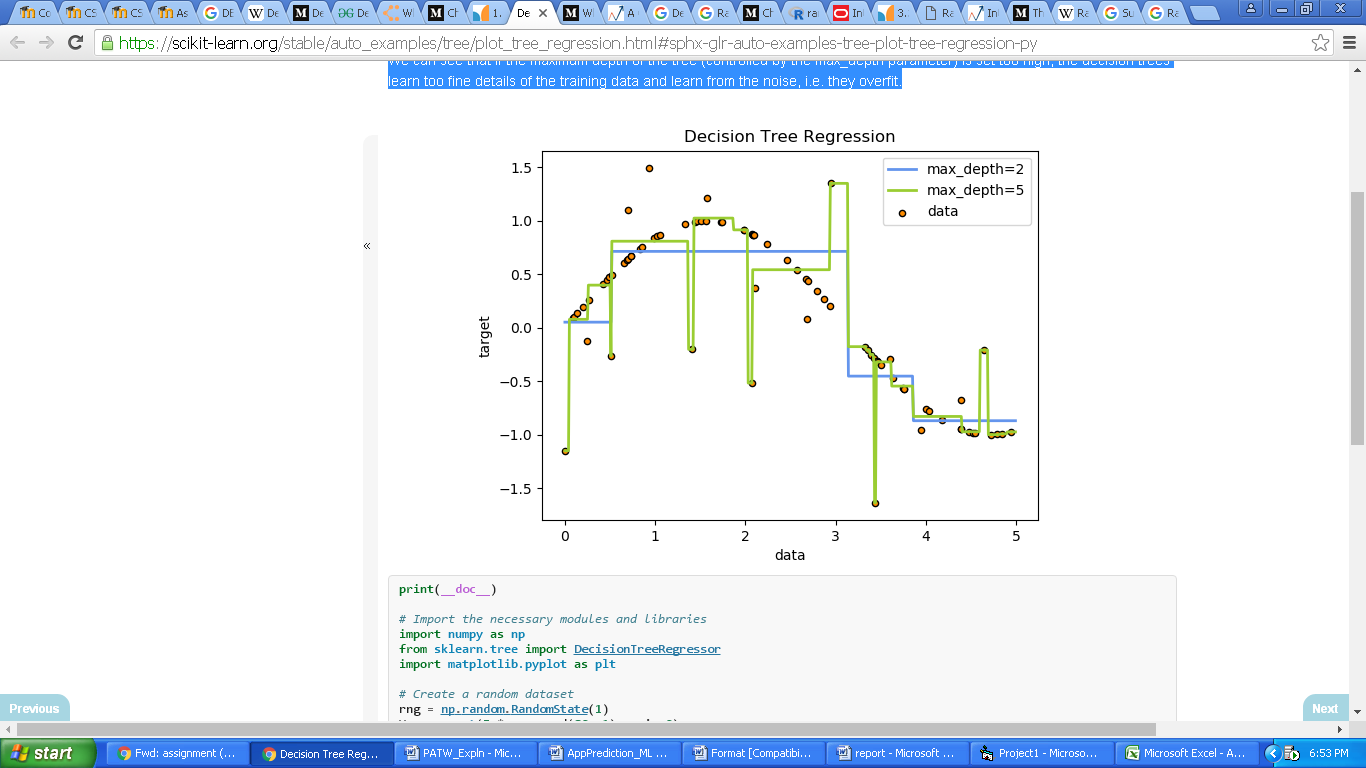
(Outlook = Sunny ^ Humidity = Normal) v (Outlook = Overcast) v (Outlook = Rain ^ Wind = Weak)

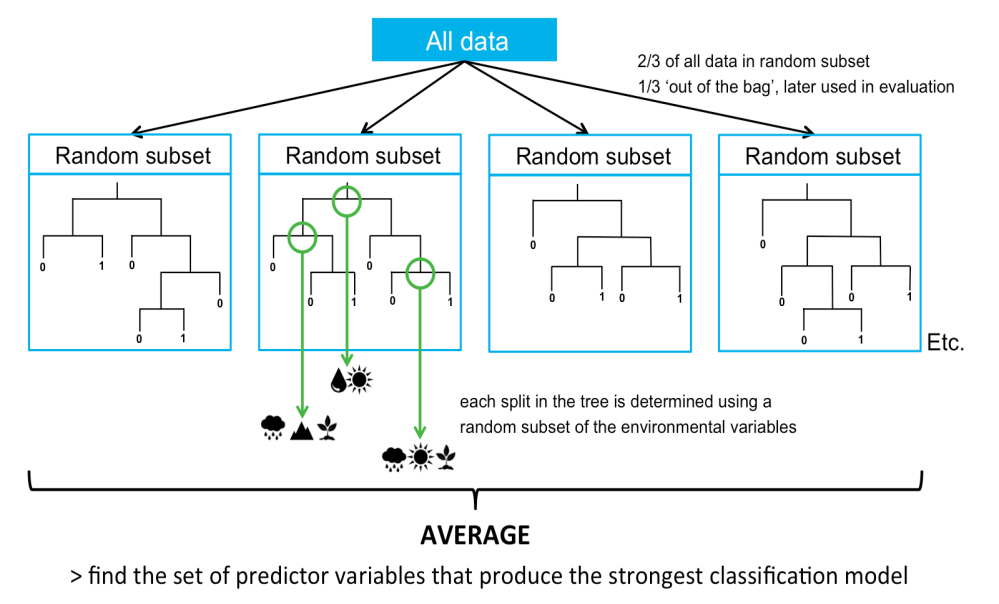
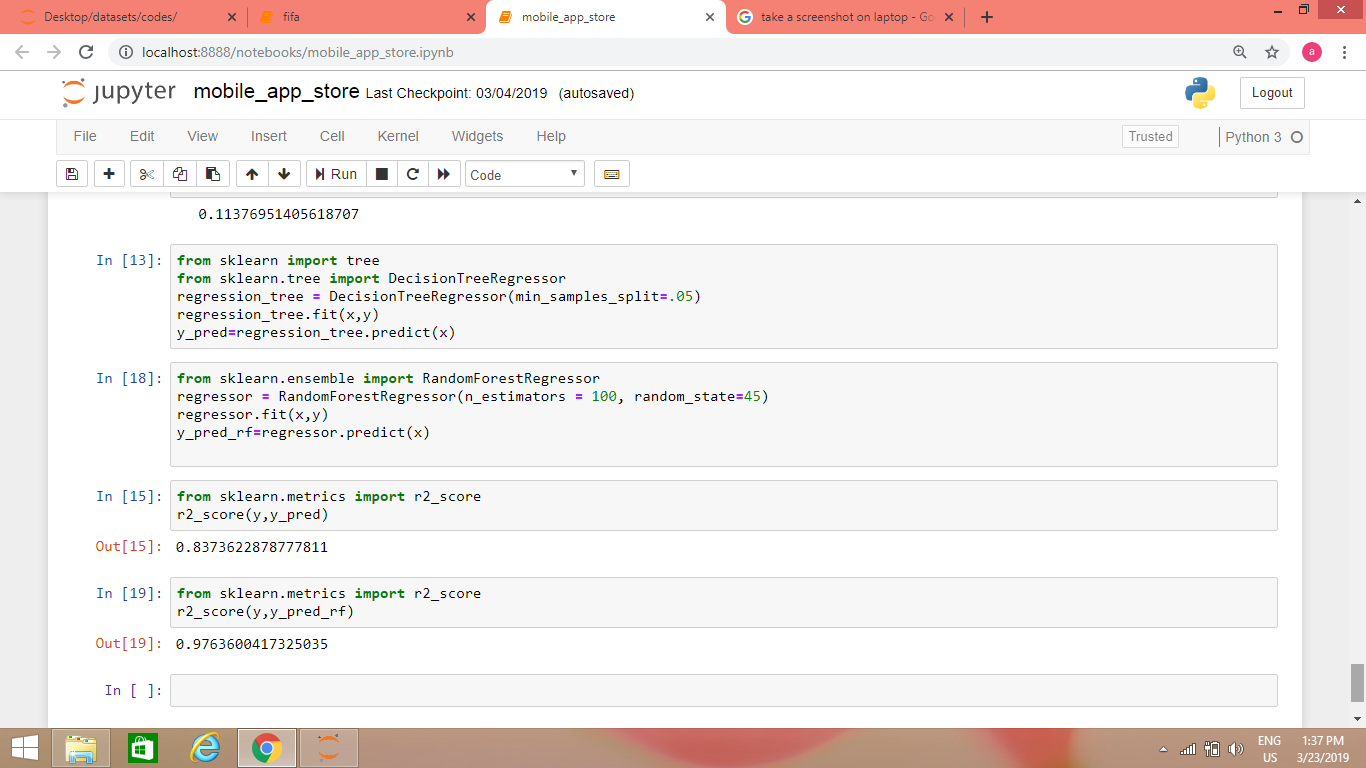
* **Strengths and Weakness of Decision Tree approach:**

**The strengths of decision tree methods are:**

* Decision trees are able to generate understandable rules.
* Decision trees perform classification without requiring much computation.
* Decision trees are able to handle both continuous and categorical variables.
* Decision trees provide a clear indication of which fields are most important for prediction or classification.

**The weaknesses of decision tree methods:**

* Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
* Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.
* Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.
* **Decision Trees (DTs)** are a non-parametric supervised learning method used for [classification](https://scikit-learn.org/stable/modules/tree.html#tree-classification) and [regression](https://scikit-learn.org/stable/modules/tree.html#tree-regression). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
* For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.
* Decision Tree Regression - A 1D regression with decision tree.
* The [decision trees](https://scikit-learn.org/stable/modules/tree.html#tree) are used to fit a sine curve with addition noisy observation. As a result, it learns local linear regressions approximating the sine curve.
* We can see that if the maximum depth of the tree (controlled by the max\_depth parameter) is set too high, the decision trees learn too fine details of the training data and learn from the noise, i.e. they over fit.

* **RANDOM FOREST CLASSIFIER**
* A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).
* It is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because its simplicity and the fact that it can be used for both classification and regression tasks. In this post, you are going to learn, how the random forest algorithm works and several other important things about it.
* Random Forest Classifier is ensemble algorithm. In next one or two posts we shall explore such algorithms. Ensembled algorithms are those which combines more than one algorithms of same or different kind for classifying objects. For example, running prediction over Naive Bayes, SVM and Decision Tree and then taking vote for final consideration of class for test object.
* Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object. Alternatively, the random forest can apply weight concept for considering the impact of result from any decision tree. Tree with high error rate are given low weight value and vice versa. This would increase the decision impact of trees with low error rate.
* Random Forests grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).
* It does not over fit. You can run as many trees as you want. It is fast. Running on a data set with 50,000 cases and 100 variables, it produced 100 trees in 11 minutes on a 800Mhz machine. For large data sets the major memory requirement is the storage of the data itself, and three integer arrays with the same dimensions as the data. If proximities are calculated, storage requirements grow as the number of cases times the number of trees.
* **Features of Random Forests:**
* It is unexcelled in accuracy among current algorithms.
* It runs efficiently on large data bases.
* It can handle thousands of input variables without variable deletion.
* It gives estimates of what variables are important in the classification.
* It generates an internal unbiased estimate of the generalization error as the forest building progresses.
* It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
* It has methods for balancing error in class population unbalanced data sets.
* Generated forests can be saved for future use on other data.
* Prototypes are computed that give information about the relation between the variables and the classification.
* It computes proximities between pairs of cases that can be used in clustering, locating outliers or (by scaling) give interesting views of the data.
* The capabilities of the above can be extended to unlabeled data, leading to unsupervised clustering, data views and outlier detection.
* It offers an experimental method for detecting variable interactions.
* ***This is the input and output code for Decision Tree Classifier and Random Forest Classifier Algorithm. Firstly, we import the DecisionTreeRegressor module from SKLearn and then fitting the training dataset to the model. Then the values of y are predicted on testing dataset. The accuracy score is found using r2\_score and it is 83.736% for Decision Tree Algorithm. Similarly, we import RandomForestRegressor module from SKLearn, fit the data and then the values are predicted. The accuracy score for Random Forest is found to be 97.636% which means that almost 98% predictions made by our model is perfectly correct.***

**COMPARING LEARNING SYSTEMS**

* ***The accuracy score of various methods/algorithms used here are:***

|  |  |
| --- | --- |
| *ALGORITHM USED* | *ACCURACY SCORE* |
| * LINEAR REGRESSION | **11.37%** |
| * DECISION TREE | **83.736%** |
| * RANDOM FOREST | **97.636%** |

* **It is seen from the above table that we get maximum accuracy in Random Forest Algorithm. Decision tree gives good result with almost 83% correct prediction. On the other hand, Linear Regression gives the worst accuracy among these three for this dataset. Thus, it is clear that Random Forest is the best algorithm with almost 98% accuracy and we are using this to algorithm in our project for prediction.**

**DISCUSSION**

Recommendation systems that model users and their interests are often used to improve various user services. Such systems are usually based on automatic prediction of user ratings of the items provided by the service. This paper presents an overview of some of the methods for automatic ratings prediction in the domain of mobile app ratings. The chosen methods are based on various approaches described in related papers. During the prediction process both the user and item features can be used. For the purpose of this paper, data was gathered from the publicly available mobile app database from the “[iTunes Search API](http://www.transtats.bhttps/developer.apple.com/library/content/documentation/AudioVideo/Conceptual/iTuneSearchAPI/SearchExamples.html#//apple_ref/doc/uid/TP40017632-CH6-SW1ts.gov/DatabaseInfo.asp?DB_ID=120&Link=0)” at the Apple Inc website. The paper encompasses the implementation of the chosen methods and their evaluation using the gathered data. The results show a comparison in different methods used and adopting the algorithm with highest accuracy which would give best prediction.

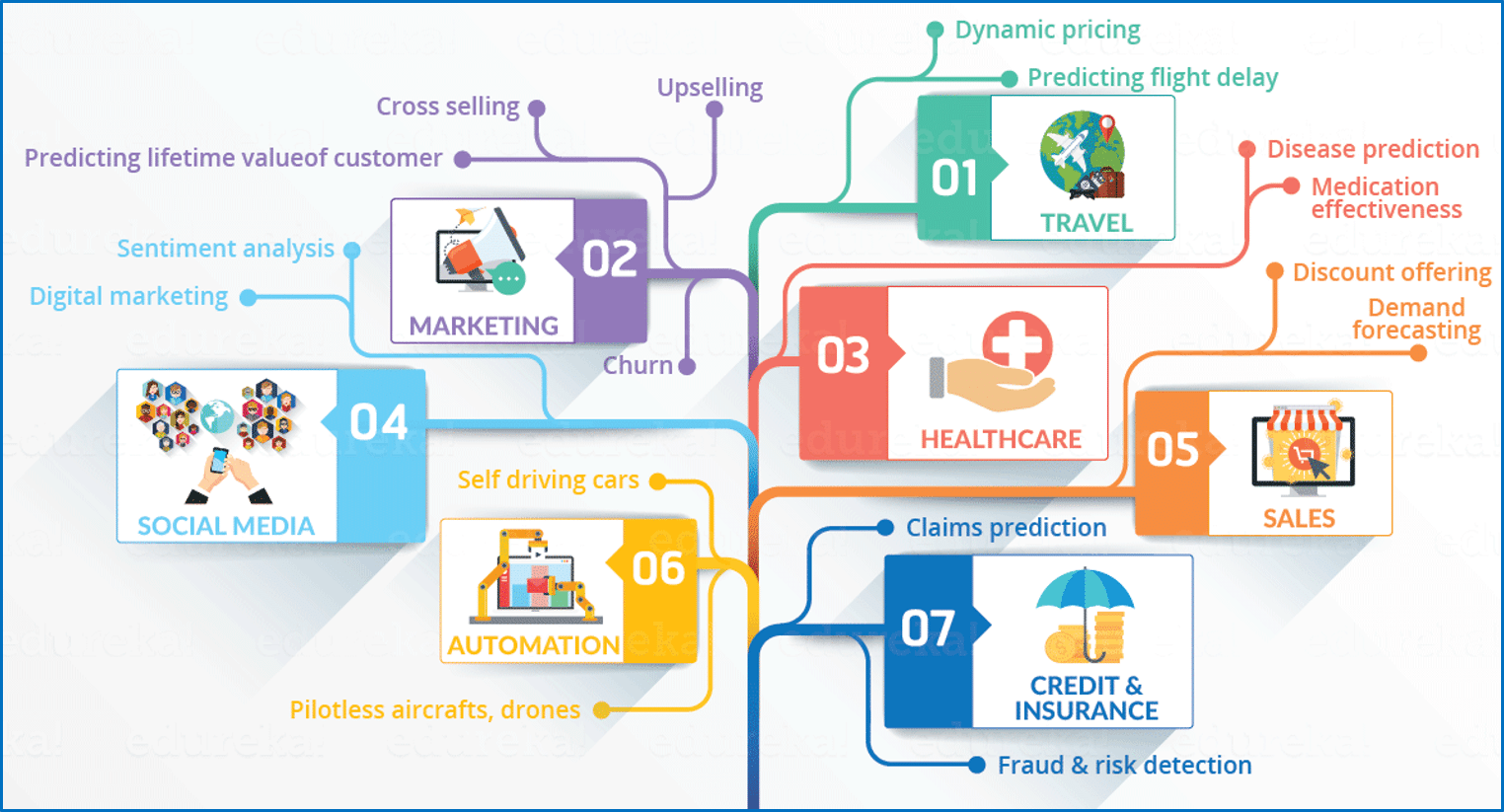
**CONCLUSION**

**As I used Random Forest, Decision Tree and Linear Regression algorithm, the best prediction result is opted by Random Forest algorithm which is 97.64%.We can say that our prediction in most cases will give the accurate result.**

**FUTURE WORKS**

Predicting user ratings for some items presents an interesting and well formed problem. Many different approaches to solving this problem were used with different methods and performance. This paper gives a comparison of a number of methods used in the prediction of mobile app ratings. The used dataset was collected from the publicly available Apple database. The results show that the random forest classifier algorithm achieved the best predictions. Future plans include more detailed testing with different datasets because the currently used dataset does not contain the required diversity of user ratings. Some of the implemented methods were not tested thoroughly enough because of their computational complexity and should be tested in detail. Other methods suggested in related work should be taken into consideration as well. A combination of classifiers and the use of weighted voting could be tested for prediction purposes. Finally, developing a new algorithm for mobile app ratings prediction is a possible option for the future.

The current offerings for machine learning and artificial intelligence are vast and feature-rich. For example, consider adding voice recognition to a mobile app, or utilize Amazon’s prediction APIs to build an intelligent product recommendation page for an ecommerce flow. It’s amazing how quickly machine learning can be adapted to fit into any industry or situation.

****

**REFERENCES**

* <https://www.python.org/>
* <https://anaconda.org/anaconda/python>
* <http://www.numpy.org/>
* <https://matplotlib.org/>
* <http://scikit-learn.org/>
* <https://pandas.pydata.org/>
* <https://pandas.pydata.org/>
* <https://ipython.org/>
* <https://kaggle.com/>
* <https://ieeexplore.ieee.org>
* <https://en.wikipedia.org/wiki/>
* https://www.geeksforgeeks.org/