Exploring BIG DATA with Python

Big data Analytics using Python

Fall18_CSC_9010_002

This is my learning, exploring big data analytics using python.

By,

Rashmi Purvachar

Villanova University, PA

Introduction to Big Data

Big data is a term used to refer to data sets that are too large or complex for traditional data-processing application software to adequately deal with. Data with many cases offer greater statistical power, while data with higher complexity may lead to a higher false discovery rate.

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It is what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

Why Is Big Data Important?
The importance of big data doesn't revolve around how much data you have, but what you do with it. You can take data from any source and analyze it to find answers that enable

1. cost reductions
2. time reductions
3. new product development and optimized offerings
4. smart decision making.

When you combine big data with high-powered analytics, you can accomplish business-related tasks such as:

- Determining root causes of failures, issues and defects in near-real time.
- Generating coupons at the point of sale based on the customer's buying habits.
- Recalculating entire risk portfolios in minutes.
- Detecting fraudulent behavior before it affects your organization.

[Source: https://www.sas.com/en_us/insights/big-data/what-is-big-data.html]

Big Data Analytics

BIG DATA ANALYTICS
Improve Operational Efficiencies
Grow Sales
Empower New "Blue Ocean" Business Models
Reduce Risk and Costs
Save Time
Self Service
Lower Complexity
Actionable Customer Insights
Reduce Churn
Predictive Analytics
Improve Customer Experience
Competitive Differentiation
Data as a Service
Data Science Services
Incubate New Ventures
Empower users of all levels with new actionable insights

Big data analytics examines large amounts of data to uncover hidden patterns, correlations and other insights. With today's technology, it's possible to analyze your data and get answers from it almost immediately – an effort that's slower and less efficient with more traditional business intelligence solutions.

Key Technologies:

Data mining: This technology helps you examine large amounts of data to discover patterns in the data – and this information can be used for further analysis to help answer complex business questions.

Hadoop: This open source software framework can store large amounts of data and run applications on clusters of commodity hardware. It has become a key technology to doing business due to the constant increase of data volumes and varieties, and its distributed computing model processes big data fast.

In-memory analytics: By analyzing data from system memory (instead of from your hard disk drive), you can derive immediate insights from your data and act on them quickly. This technology can remove data prep and analytical processing latencies to test new scenarios and create models; it's not only an easy way for organizations to stay agile and make better business decisions, it also enables them to run iterative and interactive analytics scenarios.

Predictive analytics: Predictive analytics technology uses data, statistical algorithms and machine-learning techniques to identify the likelihood of future outcomes based on historical data. It's all about providing a best assessment on what will happen in the future, so organizations can feel more confident that they're making the best possible business decision. Some of the most common applications of predictive analytics include fraud detection, risk, operations and marketing.

Text mining: With text mining technology, you can analyze text data from the web, comment fields, books and other text-based sources to uncover insights you hadn't noticed before. Text mining uses machine learning or natural language processing technology to comb through documents – emails, blogs, Twitter feeds, surveys, competitive intelligence and more – to help you analyze large amounts of information and discover new topics and term relationships.

[Source: https://www.sas.com/en_us/insights/analytics/big-data-analytics.html#dmtechnical]

Python Programming

Before we dive in more into big data analytics, we need to take a ride on Python.

Python is a powerful high-level, object-oriented programming language created by Guido van Rossum.
It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time. The syntax of the language is clean and length of the code is relatively short.

Python being a general-purpose language has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). Python is a cross-platform programming language, meaning, it runs on multiple platforms like Windows, Mac OS X, Linux,

Unix and has even been ported to the Java and .NET virtual machines. It is free and open source.

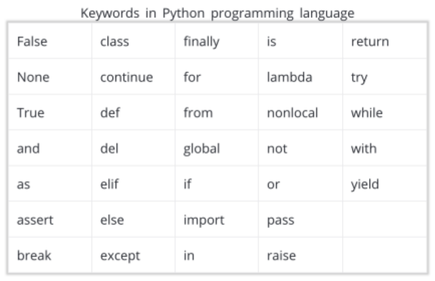
Many IDEs support python programming. I used Jupyter for my python learning.

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

[Source: https://www.programiz.com/python-programming - tutorial]

Python Keywords

Keywords are the reserved words in Python. We cannot use a keyword as variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language. In Python, keywords are case sensitive.


Keywords in Python programming language

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Python Identifiers

Identifier is the name given to entities like class, functions, variables etc. in Python. It helps differentiating one entity from another.

Rules for writing identifiers

1. Identifiers can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore (_). Names like myClass, var_1 and print_this_to_screen, all are valid example.
2. An identifier cannot start with a digit. 1variable is invalid, but variable1 is perfectly fine.
3. Keywords cannot be used as identifiers.
4. We cannot use special symbols like !, @, #, $, % etc. in our identifier.
5. Identifier can be of any length.

Python Constants and variables

Constants are usually declared and assigned on a module. Here, the module means a new file containing variables, functions etc. which is imported to main file. Inside the module, constants are written in all capital letters and underscores separating the words.

Example: Pi = 3.14

Variables do not need declaration to reserve memory space. The "variable declaration" or "variable initialization" happens automatically when we assign a value to a variable. We use the assignment operator = to assign the value to a variable.

Example: website = "Apple.com"

Data types in Python

Every value in Python has a datatype. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes. Some of the important data types are Python Numbers, List, Tuple, Strings, Set, Dictionary

Please find the work regarding the above topics in my github.

Github link 1. https://github.com/RashmiPurvachar/big-data-python-class/tree/master/Homeworks/Homework%202

Numpy and Pandas

NumPy is the fundamental package for scientific computing with Python. It contains among other things,

- 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.

Pandas is a Python package aimed to provide fast and flexible data structures designed to make working with data easy and intuitive.

Pandas provides a set of tools to make working with data simple and efficient. Pandas aims to be the most powerful and flexible open source data analysis / manipulation tool available in any language.

They support operations like,

- load an csv file and easily manipulate the data in it
- replace missing values on your data or ignore them all together
- generate a quick statistic summary of your data
- complex data filtering and slicing

[Source: https://codeburst.io/getting-started-with-data-analysis-in-python-2aaa4dbedd46, http://www.numpy.org/]

Please find the work regarding Numpy and Pandas in my github.

Github link 2. https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework3/PURVACHAR_HW3_PANDA_NUMPY_FALL2018.ipynb

Datasets from,

1. http://opendata.dc.gov/datasets
2. https://www.kaggle.com/blastchar/telco-customer-churn

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 shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Some of the types of plots that matplotlib offers are,

- Line Plot
- Multiple subplots
- Images
- Contouring and pseudocolor
- Histograms
- Paths
- Three-dimensional plotting
- Streamplot
- Ellipses
- Bar charts
- Pie charts
- Tables
- Scatter plots
- GUI widgets
- Filled curves
- Date handling
- Log plots
- Polar plots
- Legends

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

[Source: https://matplotlib.org/]

Please find the work regarding Matplotlib in my github.

Github link 3. https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework4/PURVACHAR_HW4_MATPLOTLIB_FALL2018.ipynb

Map, Filter and Reduce

These are three functions which facilitate a functional approach to programming.

- **Map** applies a function to all the items in an input list. It allows us to implement this in a much simpler and nicer way.
- **Filter** creates a list of elements for which a function returns true. It resembles a for loop but it is a built-in function and faster.
- **Reduce** is a useful function for performing some computation on a list and returning the result. It applies a rolling computation to sequential pairs of values in a list.

Few of the challenges of processing large amounts of data are,

- How to process is quickly?
- So how do we go about making the problem map so that it can be distributed computation?
- Distributed/Parallel Programming is hard

Mapreduce addresses all these challenges

- Google's computational/data manipulation model
- Elegant way to work with big data

I have used mrjob to demonstrate Map Reduce.
mrjob is the easiest route to writing Python programs that run on Hadoop. We will be able to test code locally without installing Hadoop or run it on a cluster of your choice.
Features of mrjob that make writing MapReduce jobs easier:

- Keep all MapReduce code for one job in a single class
- Easily upload and install code and data dependencies at runtime
- Switch input and output formats with a single line of code
- Automatically download and parse error logs for Python trace-backs
- Put command line filters before or after your Python code

I have used NLTK to tokenize words from many documents
NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, semantic reasoning and wrappers for industrial-strength NLP libraries.

Please find the work regarding these functions using mrjob and NLTK in my github.

Github link 4: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework5/Homework5_MapReduce_V2.ipynb

I have used TF-IDF to process textual data

TFIDF stands *for term frequency- inverse document frequency*. The TFIDF weight is used in text mining and IR. The weight is a measure used to evaluate how important a word is to a document in a collection of documents.

Please find the work regarding textual data processing using TF-IDF in my github.

Github link 5: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework5/TF%20-%20IDF%20implementation.ipynb

Linear Algebra, Statistics and Probability

Statistics and Probability in Python is understood best with an example. In my Github this is explained using 'flipping a coin' example.

This jupyter notebook also has examples on matrix creation investigation results on plotting, linear regression, and complex matrix manipulation.

It also has an illustration with a dataset - NCHS - Leading Causes of Death: United States

Dataset : https://data.cdc.gov/NCHS/NCHS-Leading-Causes-of-Death-United-States/bi63-dtpu

Please find the work regarding these in my github.

Github link 6: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework6/PURVACHAR_HW6_STATS_AND_PROB_FALL2018.ipynb

Web Scraping/Crawling in Python

Web scraping is a computer software technique of extracting information from websites. This technique mostly focuses on the transformation of unstructured data (HTML format) on the web into structured data (database or spreadsheet).

Scrapy is an application framework for crawling web sites and extracting structured data which can be used for a wide range of useful applications, like data mining, information processing or historical archival.

I have done the web Link analysis using Scrappy and website https://gizmodo.com/

My github document will provide details on how to,

- Create a crawler using the webcrawler
- create a Stochastic matrix from its resulting crawling
- provide the list of the top 5 page URLs by passing it through the Page Rank algorithm, HITS algorithm, Weighted PageRank algorithm and Moler PageRank algorithm

Please find the work regarding web crawling in my github.

Github link 7: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework7/PyLinkAnalysis/PURVACHAR_HW7_LinkAnalysis_Fall2018.ipynb

Data Streaming

There are tools and concepts in computing that are very powerful but potentially confusing to novices. One such concept is **data streaming** (aka lazy evaluation), which can be realized neatly and natively in Python.

Data streaming can be used to capture real-time data from social media websites or any other websites. The streamed data is then mined to pull out the useful information required. Sentiment analysis and similar methods are used on the streamed data for analysis which will in turn help businesses make proper decisions.

Data Streaming can be done in many methods. One of the top methods being usage of Bloom filters.

A Bloom filter is a space-efficient probabilistic data structure, conceived by Burton Howard Bloom in 1970, that is used to test whether an element is a member of a set. False positive matches are possible, but false negatives are not – in other words, a query returns either "possibly in set" or "definitely not in set". Elements can be added to the set, but not removed (though this can be addressed with a "counting" filter); the more elements that are added to the set, the larger the probability of false positives.

Please find the work regarding data streaming in my github.

Github link 8: https://github.com/RashmiPurvachar/big-data-python-class/tree/master/Homeworks/Data%20Streaming

I have used my twitter account to capture and save real time data.

This needs a developer account to be created on https://developer.twitter.com/content/developer-twitter/en.html

An application to be created to get access tokens that are required for authentication.

Algorithms in big data Analytics

Association Rule Mining

Association rule mining is a technique to identify underlying relations between different items. Take an example of a Super Market where customers can buy variety of items. Usually, there is a pattern in what the customers buy. For instance, mothers with babies buy baby products such as milk and diapers.

Association rules are a set of rules derived from a database, that can help determining relationship among variables in a large transactional database.

Apriori Algorithm for Association Rule Mining:

Different statistical algorithms have been developed to implement association rule mining, and Apriori is one such algorithm.

The steps in implementing Apriori Algorithm are:

1. Create a frequency table of all items that occur in all transactions.
2. Select only those (significant) items - for which the support is greater than threshold (50%)
3. Create possible pairs of all items (remember AB is same as BA)
4. Select itemsets that are only significant (support > threshold)
5. Create tiplets using another rule, called self-join. It says, from the item pairs AB, AC, BC, BD, we look for pairs with identical first letter. So, we from AB, AC we get ABC. From BC, BD we get BCD.
6. Find frequency of the new triplet pairs, and select only those pairs where the support of the new itemset (ABC or BCD) is greater than the threshold.
7. If we get 2 pairs of significant triplets, combine and form groups of 4, repeat the threshold process, and continue.
8. Continue till the frequency after grouping is less than threshold support.

Please find the tutorial I have created on Apriori algorithm implementation in my github.

I have used store dataset for this implementation.

Dataset : https://drive.google.com/file/d/1y5DYnodGoSbC22xowBq2d4po6h1JxcTQ/view

Github link 9: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework8/PURVACHAR_HW8_AprioriAlgo_Fall2018.ipynb

ECLAT Algorithm for Association Rule Mining:

ECLAT stands for Equivalence CLAss Transformation

This is an algorithm for Frequent Pattern Mining based on Depth-First Search traversal of the itemset Lattice

- a DFS traversal of the prefix tree than lattice
- Branch and Bound method is used for stopping

ECLAT Steps:

1. Get tidlist for each item (DB scan)
2. Tidlist of {a} is exactly the list of transactions containing {a}
3. Intersect tidlist of {a} with the tidlists of all other items, resulting in tidlists of {a,b}, {a,c}, {a,d}, … = {a}-conditional database (if {a} removed)
4. Repeat from 1 on {a}-conditional database
5. Repeat for all other items

Pros

- Depth-first search reduces memory requirements.
- Usually (considerably) faster than Apriori, ECLAT is used to improve the efficiency of Apriori.
- No need to scan the database to find the support of (k+1) itemsets, for k>=1.

Cons

- The TID-sets can be quite long, hence expensive to manipulate.

Please find the tutorial I have created on ECLAT v/s Apriori algorithm implementation in my github.

I have used store dataset for this implementation.

Dataset: https://www.kaggle.com/shwetabh123/basketballoptimisation

Github link 9: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework8/PURVACHAR_HW8_APRIORI_ECLAT_Fall2018.ipynb

Clustering

Clustering as a general is a technique that most of us do in our lives. For instance, sharing a same table in a restaurant, or in a super market where similar products are placed together or nearby location.

K-Means Clustering Algorithm¶

K-Means is a clustering algorithm whose main goal is to group similar elements or data points/items into a cluster. 'K' here represents the number of clusters/groups. The algorithm runs iteratively to assign each data points to one of the 'K' groups based on the features that are known or provided.

A K-Means algorithm can be applied to numeric or continuous data with smaller number of dimension. It can be also applied to any scenario where a groups of similar things has to be pulled off from randomly distributed collection of data.

For example, document classification. Clusters of documents can be created in multiple categories based tags, topics and contents of the document. This is the very standard classification problem and K-Means algorithm is one of the highly suitable solution that serves the purpose.

Please find the tutorial I have created on K-Means clustering algorithm implementation in my github.

I have used iris dataset for this implementation.

Github link 10: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework8/PURVACHAR_HW8_ClusteringAlgo_Fall2018.ipynb

Classification

Classification is a process of diving a data set into different cat of groups by adding labels. Generally, the process goes as, taking the data, analyzing it and based on some conditions the data set is divided into various categories.

Classification is necessary to perform prediction analytics on the data for better organization and access. For instance, Gmail recognizes an incoming email and categorizes as spam, social, promotions or into general inbox.

Types of Classification

There are several types of classifications which are currently used based on the need. Listed here are few among them. Decision Tree, Random Forest, Naive Bayes, KNN.

Decision Tree

Decision tree is a graphical representation of all possible solutions to a decision, where decision outcomes are based on some conditions. Decision tree enables easy understanding of the path from problem statement to solution.

Please find the tutorial I have created on Decision Tree algorithm implementation in my github.

I have created a small dataset for this implementation.

Github link 11: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework8/PURVACHAR_HW8_ML_DecisionTreeAlgo_Fall2018.ipynb

Dimensionality Reduction

In the field of machine learning, it is useful to apply a process called dimensionality reduction to highly dimensional data. The purpose of this process is to reduce the number of features under consideration, where each feature is a dimension that partly represents the objects.

As more features are added, the data becomes very sparse and analysis suffers from the curse of dimensionality. Additionally, it is easier to process smaller data sets.

Dimensionality reduction can be executed using two different methods:

- Selecting from the existing features (feature selection)
- Extracting new features by combining the existing features (feature extraction)

The main technique for feature extraction is the Principle Component Analysis (PCA). PCA guarantees finding the best linear transformation that reduces the number of dimensions with a minimum loss of information. Sometimes the information that was lost is regarded as noise – information that does not represent the phenomena we are trying to model, but is rather a side effect of some usually unknown processes.

Principal component analysis is a fast and flexible unsupervised method for dimensionality reduction in data. Its behavior is easiest to visualize by looking at a two-dimensional dataset.

Using PCA for dimensionality reduction involves zeroing out one or more of the smallest principal components, resulting in a lower-dimensional projection of the data that preserves the maximal data variance.

Please find the tutorial I have created on Dimensionality reduction and Clustering implementation in my github.

I have illustrated,

- Dimensionality reduction using sklearn PCA
- Hierarchical with sklearn Agglomerative Clustering

I have the dataset from https://archive.ics.uci.edu/ml/datasets/seeds

Github link 12: https://github.com/RashmiPurvachar/big-data-python-class/tree/master/Homeworks/Homework9

Recommender Systems

A **Recommender System** is one of the most famous applications of data science and machine learning.

The main goal is to provide the most relevant information to a user by discovering patterns in a dataset.

A typical recommendation engine processes data through the following four phases namely,

- Data Collection
- Data Storage
- Data Analysis
- Data filtering

Important types of recommendations engines,

- Collaborative filtering
- Content-Based Filtering
- Hybrid Recommendation Systems

Please find the tutorial I have created on Recommender Systems in my github.

It has more details about,

1. Recommender systems
2. How Does a Recommendation Engine work?
3. Different types of recommendations
4. Applications of Recommendation Engines
5. Advantages of Recommendation systems
6. Use case - Movie Recommender System

Dataset source : https://grouplens.org/datasets/movielens/latest/ ("ml-latest-small.zip" file)

Github link 13: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework10/PURVACHAR_HW10_RecommenderSystem_Fall2018.ipynb

Big data algorithms

Naïve Bayes

"Bayes' theorem is named after Reverend Thomas Bayes, who first provided an equation that allows new evidence to update beliefs in his 'An Essay towards solving a Problem in the Doctrine of Chances' (1763).

It was further developed by Pierre-Simon Laplace and then Sir Harold Jeffreys put Bayes' algorithm and Laplace's formulation on an axiomatic basis. Jeffreys wrote that Bayes' theorem "is to the theory of probability what the Pythagorean theorem is to geometry"."
Bayes Theorem gives the conditional probability of event an A given another event B has occured.

"Naïve Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "Naïve" assumption of conditional independence between every pair of features given the value of the class variable."

Classification and Regression are the types of supervised ML techniques and Naïve Bayes algorithm is one of the powerful tools in this ML library.

Pros: Naïve Bayes learners and classifiers can be extremely fast compared to more sophisticated methods.

Cons: It is known to be a bad estimator, so the probability outputs from predict_proba cannot be taken too seriously.

Please find the tutorial I have created on Naïve Bayes in my github.

It has more details about,

1. Bayes theorem introduction
2. What is Naïve Bayes?
3. Where do we need Naïve Bayes?
4. Understanding Naïve Bayes Classifier
5. Advantages of Naïve Bayes Classifier
6. Use case - Text Classification using Naïve Bayes Classifier

Dataset used is - 20 newsgroups dataset.
Ref : https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html

Github link 14: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework10/PURVACHAR_HW10_NaiveBayes_Fall2018.ipynb

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The Decision of the majority of the trees is chosen by the random forest as the final decision.

Classification and Regression are the types of supervised ML techniques and Random Forest algorithm is one of the powerful tools in this ML library.

Advantages to use random Forest over other algorithms. The top benefits are,

No overfitting

- Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.
- The testing performance of this does not decrease (due to overfitting) as the number of trees increases.
- Takes less time for training data.

High Accuracy

- It can run efficiently handling large data providing highly accurate predictions.

Estimates Missing data

- Missing data problem can occur when the input data is coming from multiple sources and different statistics are used to analyze this data.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

Please find the tutorial I have created on Random Forest in my github.

It has more details about,

1. What is Machine Learning?
2. What is Random Forest?
3. Why Random Forest?
4. How does a Random Forest work?
5. What are the applications of Random Forest?
6. Use case - Iris Flower Analysis

Dataset used is - Iris dataset.

Github link 15: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/Homework10/PURVACHAR_HW10_RandomForest_Fall2018.ipynb

Machine Learning

Machine learning (ML) is the study of algorithms and mathematical models that computer systems use to progressively improve their performance on a specific task.
Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.

Major categories of ML

- Supervised Learning: The correct classes of the training data are known.
- Unsupervised Learning: The correct classes of the training data are not known.
- Reinforcement Learning: Allows the machine or software agent to learn its behavior based on feedback from the environment. This behavior can be learnt once and for all, or keep on adapting as time goes by.

Machine Learning Techniques

Classification: is a kind of problem wherein the outputs are categorical in nature, like Yes or No, True or False, 0 or 1.

- This predicts classes from observations.
- Classification algorithms: KNN, Naive Bayes, Decision Tree and Random Forest.

Clustering: is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other.

- Group observations into “meaningful” groups.
- Clustering algorithms: K-Means, Hierarchical clustering

Regression(prediction): is a measure of the relation between the mean value of one variable (e.g. output) and corresponding values of other variables (e.g. time and cost).

- Regression analysis is a statistical process for estimating the relationships among variables.
- Regression means to predict the output value using training data.
- Predict value from observations.
- Regression algorithms: Logistic regression, Linear and Polynomial Regression, Regression Trees.

Tensor flow is one of the top Deep Learning libraries, open source, developed by Google Brain Team and written in Python, C++ and CUDA.
Tensor Flow bundles together a slew of machine learning and deep learning/neural networking models and algorithms and makes them useful by way of a common metaphor.

Tensor is a generalization of vectors and matrices of potentially higher dimensions. Arrays of data with different dimensions and ranks that are fed as input to the neural network called Tensors.

Please find the tutorial I have created on Machine Learning illustrating TensorFlow in my github.

Dataset used is - Boston.
Ref: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html

Github link 16: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/MachineLearning/PURVACHAR_ML_TensorFlow.ipynb

Deep Learning

Deep Learning is a subset of Machine Learning that has networks which are capable of learning from data that is unstructured or unlabeled and works similar to the functioning of a human brain.

Unlike in Machine Learning, the data is more complicated and unstructured like an image or video files. One of the core components in deep learning is Neural network. A neural network has one to many hidden layers between an input and an output layer. Anything more than one hidden layer is known as Deep Neural Network.

Deep Learning learns from the data that is unstructured and uses complex algorithms to train a neural net.

Input Layer - It accepts large amounts of data as an input to build the data. (Images, Pixel values of images, Video files etc.,). This data will be passed onto the hidden layers.

Hidden layer - These process the data by performing complex operations and computations and carries out feature extraction and then passes the values to the output layer.

Output layer - This generates the predicted output by applying suitable activation functions.

Please find the tutorial I have created on Deep Learning in my github.

It has more details about,

1. Deep Learning
2. Benefits of Deep Learning
3. Neural networks
4. Types of Neural networks
5. Applications of Deep Learning
6. Introduction to TensorFlow
7. Use case implementation using TensorFlow

Dataset used is - MNIST. This database is an integral part of TensorFlow, we can directly import the input data from tensorflow.examples.tutorials.mnist, and so we will not need any file download for the dataset.

Github link 17: https://github.com/RashmiPurvachar/big-data-python-class/blob/master/Homeworks/MachineLearning/PURVACHAR_DeepLearning_Tutorial.ipynb

Project Submissions

Domain Presentation: Big Data in Retail and E-Commerce

Big data is helping retailers to understand their prospects on a deeper level and with a host of metrics including social media preferences, browsing behaviors, devices preferences, geographical demographics and much more readily available, brands are branching out in more meaningful ways than ever before.

Big data offers in-depth information about the people your brand is targeting and it's changing the face of the retail world in a colossal way.

Please find the github link for the submitted documents.

Github link 18: https://github.com/RashmiPurvachar/big-data-python-class/tree/master/Homeworks/Domain%20Presentation

Final Presentation: Recommender systems

A **Recommender System** is one of the most famous applications of data science and machine learning.

The main goal is to provide the most relevant information to a user by discovering patterns in a dataset.

A typical recommendation engine processes data through the following four phases namely,

- Data Collection
- Data Storage
- Data Analysis
- Data filtering

Important types of recommendations engines,

- Collaborative filtering
- Content-Based Filtering
- Hybrid Recommendation Systems

Please find the github link for the submitted documents.

Github link 19: https://github.com/RashmiPurvachar/big-data-python-class/tree/master/Homeworks/Final%20Presentation