PROFESSIONAL PROJECTS

Title: Ingestion, Data Analysis and Visualization of Cardiovascular Patient Data **June 2017- February 2018**

The project involves creating an ETL pipeline through which data is passed to Docker through web application, CSV upload or MySQL scripts, where encryption and anonymity are taken care of and data is ingested in MySQL database. From MySQL database the data is passed to R and PySpark Algorithms. A second ETL is created which involves ingesting the output data from algorithms to Druid. The whole flow is orchestrated using Apache Airflow. The data from druid is then displayed on a web application written in Play Framework. Various charts are created using Apache Superset which is tightly integrated with Druid and is displayed on web application. My role in this project is to develop an end to end flow of the project. I have worked over writing partner specific ETL pipeline in Scala and Spark which involves ingesting the data, anonymizing and transforming it and store it in MySQL database. I have also worked on creating a Docker with distributed containers that have scalable master slave architecture (increase or decrease in number of slave nodes without rebuilding the Docker image). I have added Hadoop, Apache Spark, MySQL, Apache Airflow, R, Python, Java, Scala and Zeppelin Notebook in the docker. The data is passed to various R and PySpark based machine learning data analysis algorithms and output is written in form of CSV. The whole system is deployed in above created docker and orchestrated through Apache Airflow. I was responsible for developing a Docker with Druid, Superset, Zookeeper, Java, Python installed and deploying it on AWS EC2 machine. I have written a micro service in Play Framework which acts as a middle-ware between UI and Apache Airflow and involves in functionalities such as triggering the direct acyclic graph of jobs, collecting the data from UI, passing the data to the DAG, checking status of DAG whether it is running, success or failed by making a call to Airflow MySQL meta-database and if successful taking the output from dag and passing it to UI. I have worked on the second ETL written in scala and spark, which involves transforming the data to make it able to ingest to druid. Besides working over the core development, I was also responsible for designing and creating the environment for deploying the application on AWS EC2 instances and working on AWS VPC, S3 and RDS while also adding the functionality of continuous integration using Jenkins and automating the deployment process with the help of Salt-Stack.
Environment: Platform & Technologies- Python, Scala, Play Framework, Apache Hadoop, Apache Spark, Docker, R, Apache Airflow, Druid, Apache Superset, MySQL, Saltstack, Jenkins, AWS, SBT, Front End Technologies

Title: Recommendation System and Chatbot **December 2016 – May 2017**

The project started with creating a knowledge base by crawling the data using Apache Nutch web crawler. This involved writing an HtmlParseFilter plugin in which the useful data was extracted with the help of regular expressions and saving the data to a file. Then, the data was cleaned using Python Scripts and output was generated in form of a CSV and multiple Jsons, in which a separate Json was generated for each row of CSV. The CSV is used to insert the data in Neo4J while the Jsons were used for inserting the data to Elasticsearch. In case of the *Recommendation System*, when the user enters the query in the search box, the UI made a GET call to the end point exposed by the middleware and also passed user query

as a query parameter. This triggered a function that made a query to the elastic search and passed the required response to the UI. For Chat-Bot, hierarchical Jsons were generated from Neo4J and passed to API.AI in order to train the chat-bot. The whole system was then deployed on AWS.
Environment: Platform and Technologies- Java, Python, Neo4J, Nutch 1.12, Play Framework, Elasticsearch, API.AI, AWS, ANT, SBT, Maven, Front End Technologies.

Title: Contribution to Exadatum Products November 2016 – February 2018
1. Dockerized the company product XStream in a docker with Hadoop, Java, Scala, Python, Spark, Zookeeper, Kafka, SQL, Druid, Superset, Hive, Maven and Redis installed.
2. Architected and developed the automatic AWS instance management system using Python and Boto3 for Company product XInterview.
3. Given company level training sessions on Docker and Pycharm Remote Debugging.

ACADEMIC PROJECTS

Title: Query-Focused Multi-Document Summarization August 2015 – May 2016
Worked on a project that involved taking multiple documents and query as input and pre-processing both document and query by performing stemming and stopword removal with the help of Stanford CoreNLP library. The documents were broken into sentences and query was expanded by generating synset with the help of WordNet library. Multiple documents and queries were represented as vectors in vector space model. The cosine similarity of each sentence with the query was calculated and on the basis of calculation, the more similar sentences were selected and the summary was generated as output.
Environment: Java, Stanford CoreNLP, WordNet, Rida-WordNet, Lucene, Maven

Title: Text Based Plagiarism Detection January 2015 – May 2015
Developed an efficient technique to detect plagiarism in text-based documents. This project introduced a mechanism that combined the functionality of substring matching and keyword similarity to provide more efficient results. When there was a huge number of documents to which plagiarized document is to be compared, it would require a lot of time; Cluster of documents was created to make this task less time consuming that contains only those documents having a high F-Score. The F-Score was calculated using the lengths of both plagiarized document and the reference document and their longest common subsequence. A research paper was published in an international journal based on this project in April 2016.
Environment: Python

Title: SpeedAhead.com August 2014–December 2014
Designed SpeedAhead.com - a dynamic website that displays different cars and their catalog. This website aided in connecting the user to the car sellers and book the cars directly online. Some of the functionalities included are adding and removing users/sellers/cars, view catalog, users, admin and seller login.
Environment: PHP, HTML, CSS, Bootstrap, JavaScript, MySQL, Wampp Server

Title: 8085 Microprocessor Simulator **January 2014 – April 2014**

The project was developed in C language. This project involved simulating registers such as general purpose registers, Accumulator, and Program Counter, main memory and also simulating various instructions related to data transfer, arithmetic and logical operations, and branching.

Title: Vehicle Management System **August 2013 – December 2013**

The system was developed in C++. The functionalities include add and remove users, add and remove sellers, view catalog, users, admin, seller login, exporting the details in a txt file.