KUSHAGR JOLLY

kushagrjolly15@gmail.com | +1 (682)-331-5254 www.linkedin.com/in/kjolly

TECHNICAL SKILLS

- **Languages:** Python, Java, R, C++, C, PHP, C#, Android, Dot.NET, JavaScript
- **Machine Learning:** OpenCV, NLTK, TensorFlow, Keras, PyTorch, Scikit-Learn, Scipy, Pandas, NumPy
- **Databases:** MySQL, Oracle, Teradata, MSSQL, PostgreSQL, MongoDB, Greenplum.
- **Web Technologies:** Flask, Django, PHP, XHTML, CSS, Bootstrap, JQuery
- **Analytical Tools**: Tableau, BobJ, Power BI, MS Excel.
- **Tools:** Amazon Web Services (AWS), VS Code, PyCharm, Anaconda, IntelliJ, Eclipse, Git, Linux, Putty, Bamboo, Bitbucket, JIRA, Docker, Confluence, Postman, Spring Boot, Maven, SonarQube, APIGEE, Splunk.

EDUCATION

- MS in Computer Software Engineering. GPA: 3.73/4.0. Arizona State University. *May 2021*
- Bachelor of Technology in Computer Science Engineering. Shiv Nadar University, India. *August 2013- May 2017*

CERTIFICATIONS: IBM Data Science, Applied Data Science with Python, Machine Learning, Deep Learning Specialization.

WORK EXPERIENCE

Complex Adaptive Systems, Tempe AZ – *Research Aide* *June 2020 – Present*

- Helped align journal articles into different folders to organize them into sub-categories.
- Developed an **NLP based OCR model** to extract text from documents aiding a professor in his research.
- **Utilized**: Python, NLP, MySQL, Flask, JavaScript, HTML, CSS, Bootstrap

Dell Technologies, India - *Software Engineer 1* *July 2017- July 2019*

- Developed an end to end **software design architecture** and implementation following **SDLC** including defining requirements, prototyping, designing, coding, testing, deploying and maintaining software for the Customer Account Management System.
- Implemented **Sales-Prediction Tool** for pre-sales team; Responsible for end-to-end **data science** project life cycle and supervised all phases including data cleaning, data extraction and data visualization with big data.
- Built the **Kafka** streaming pipelines to migrate data from upstream into AVRO files and loading it into **Kafka** topics and eventually to the destination (Teradata); which increased team efficiency by 16%.
- Processed and analyzed data collected at data staging layer by writing stored procedures and storing data as facts and dimensions in data warehouse built on **Teradata**.
- **Awards**: Dell Quarter Award in Q3FY17, Q2FY18.

Indian Agricultural Statistical Research Institute (IASRI), India - *Research Assistant* *May 2015*

- Developed an application to calculate the **leaf area of the plant using image recognition** and classification;
- **Utilized**: Android, MySQL server, Python

PROJECTS

- **Sentiment Analysis**
  - A Yelp review dataset was cleaned and tokenized using **Natural Language Toolkit (NLTK)** and stop words were not removed considering the fact, each word or punctuation can impact the meaning of the sentence.
  - Various method was used like Cross-Validation, Neural Network, Support Vector Machines, Logistic Regression etc. to train the model and identify the best algorithm for building the model.
  - **Achieved Accuracy of 87% on test data;** Deployed the models **Google Cloud ML Engine**
  - **Utilized**: Python, NumPy, Pandas, Matplotlib, Sklearn, NLTK.
- **Banking System**
  - Built Kafka producers to perform batch processing and batch-loading into **Kafka** topics, for structured/semi-structured data.
  - Utilized: **Python, Kafka**, MySQL, MS-SQL Server, HTML, CSS, JavaScript, Bootstrap
- **Sales Prediction Model**
  - Developed a web application using Django - Rest Framework that predicts the sales price of server rack using Random forest, kNN, rule-based implementation.
  - Created a production-ready REST API with Python and Django-REST-Framework, with endpoints providing the selling price of racks; Designed a scalable relational database using Teradata;
  - Utilized: **Python, Django, Django-REST-Framework, Teradata** Database, JavaScript, HTML, CSS, Bootstrap
- **Complex Spatial Querying**
  - Process queries entered by the end client; mCK query aims to find the spatially closest tuples which match m user-specified keywords using algorithms like key closest descriptor, kNN, point in polygon.
  - Utilized: Android, MySQL DB, Google Maps API