




Happy Sisodia

(864) 633 – 7003 • happy.a.sisodia@gmail.com • 102 Calhoun St, Clemson, SC-29631 •  •  • 

SUMMARY

An experienced data scientist with expertise in analyzing and synthesizing large data sets for critical business decisions. Strong experience in Advanced Programming and passion for exploring patterns and relationships in large volumes of data to solve complex problems using advanced statistical models.

EDUCATION

Master of Science in Computer Science | Clemson University

Jan 2019 – Dec 2020

Relevant Coursework: Data mining, Artificial Neural Network, Data Analysis, Applied Data Science, Artificial Intelligence, DBMS

Bachelor of Engineering in Computer Science | Gujarat Technological University

Aug 2011 – Jul 2015

PROFESSIONAL EXPERIENCE

Thirdware Solution, Mumbai, India

Sep 2015 – Dec 2017

Data Business Analyst

- Led a team of interns to develop a Goods and Services Tax (GST) module using QAD which identified different tax codes.
- Automated the Global Requisition System (GRS) by optimizing the approvals process which led to a saving in time of over 30%.
- Performed requirement gathering by liaising with cross-functional clients in manufacturing and distribution departments.
- Created batch programs to automate processes such as invoice creation, generating sales orders and ETL.
- Created documents such as Technical specification, Test case and System documents that was used by cross-functional teams.
- Translated functional requirements to technical teams and presented module output demos to clients on a weekly basis.

TECHNICAL SKILLS

Methodology and statistical knowledge: Neural Networks, Naïve Bayesian, Logistic regression, SVM, K-Means, Quantitative analysis, Statistical modeling, Agile software Development

Programming: Python, R, SQL, LaTeX

Tools: MS office, Tableau, QAD, GitHub

ACADEMIC PROJECTS

CNN for Image Classification || [GitHub Link](#)

- Implemented a Convolution Neural Network (CNN) model with Keras to predict the classification of types of images using a [CIFAR-10](#) dataset consisting of 60K images divided into 10 balanced classes.
- Performed data Normalization and One Hot Encoding, and used ReLu to prevent the exploding gradient problem during back propagation.
- Procured a model accuracy of **75%** after training the model for 100 epochs with a learning rate of 0.01 using RMSprop optimizer.

Sparse K-means Classification || [GitHub Link](#)

- Analyzed and compared the performance of K-Means on the data matrix using 3 separate sets of data.
- Added the Random Projection and matrix sparsification pre-processing steps to increase the performance of K-means algorithm.
- Used K-Means++ initialization method algorithm to obtain clusters which led to an increase in performance by 60%

Neural Network to test Injury severity || [GitHub Link](#)

- Applied a Feed Forward Neural Network to a [FARS](#) dataset to examine relationships between driver injury severity, driver, vehicle and roadway condition.
- Selected, trained and evaluated low multi-class classification systems having high problem domain complexity with an imbalanced dataset.
- Created a MLP from scratch using only Pandas libraries in python with stochastic gradient descent which provided an accuracy of 46% for the multi-class classification and 60% for the binary classification.

Multi Label Categorization of Construction Projects || [GitHub Link](#)

- Successfully extracted and labelled information such as construction, design, operation and maintenance from construction contracts and project requirement documents using supervised learning in python.
- Performed operations such as punctuation removal, tokenization using white space, stemming and lemmatization to pre-process the data.
- Used 'Bag of Words' to turn words into vectors and trained learning models such as Naïve-Bayesian, SVM, logistic regression and feed forward neural network.
- Performed data cleaning using methods such as Word2Vec, Doc2vec and applied multi classification using SVM which provided 90% accuracy.

FCN For Image Segmentation || [GitHub Link](#)

- Trained a machine learning model to segment an image into multiple objects built using a dataset of ~12K images and 20 classifications.
- Performed pre-processing to resize the images and appended a model to a Fully Connected Network(FCN) with a base of VGG16 and Keras library.
- Trained the model for 100 epochs with the learning rate of 0.01 and used stochastic gradient descent to obtain an accuracy of 84.5 %