Arthi Murali

9940627763 · meetarthi@gmail.com · meetarthi.com
linkedin.com/in/meetarthi · github.com/meetarthi · medium.com/@meetarthi · kaggle.com/arthimurali

---

As a passionate and skilled Data Engineer with knowledge of Data Science, I have honed my abilities to extract valuable insights from different data models. With experience in developing and optimizing data pipelines, I am eager to utilize my skills in constructing data infrastructure that enables data-driven decision making. I am enthusiastic to take on new challenges, expand my skill sets, and contribute to the success of organizations in the field of data engineering.

SKILLS

Languages

Python, PERL, HTML, CSS, Javascript, and Data Structures (Python).

Data Engineering

- **Pandas** - Data I/O (Input/Output), Cleaning, Indexing and selecting data, handling missing values, filters, higher-order functions (Lambda, map, filter, and reduce), Pivot table, Grouping and sorting, Joins, Combining Datasets.
- **SQL** - DDL (Create,Drop,Alter,Truncate,Rename) , DQL (Select) , DML (Insert, Update , Delete).
- **MongoDB(NoSQL)** - Mongo CRUD operations (create, read, update, delete), MongoDB compass (GUI for analyzing data in MongoDB), Aggregate pipeline, Embedded documents, BLOB (Binary Large Object), Regular expressions.
- **Cloud computing(AWS)** - S3, EC2, Lambda, RDS, DynamoDB, Glue, Athena, CloudWatch, and IAM.
- **Operating System** - Linux (Ubuntu), Windows 10, Mac OS.
- **Hadoop ecosystem** - HDFS (Hadoop Distributed File System), MapReduce, Spark, HBase.
- **Airflow** - DAGs (Directed Acyclic Graphs), Operators, Scheduling and Triggers, Monitoring and Logging, Error Handling.
- **Data Processing** - Batch data processing using Pyspark on file formats like CSV, JSON, and Parquet files.
- **Monitoring/ Logging** - Prometheus, Grafana.
- **Platforms and Tools** - App Hosting Platform (Streamlit Community Cloud and HuggingFace spaces), MongoDB Atlas, Sublime, Visual Studio, Google Colab, Gradio, Docker Desktop, and XAMPP.

Data Science

- **Statistics** - Variation and Standard deviation, Data distribution, Covariance and correlation, conditional probability, Bayes theorem, probability density function, A/B testing
- **Regression** - Linear, polynomial, multiple and Logistic regression

- **Data visualization and analysis** - Numpy, Matplotlib, Seaborn and Plotly.
- **ML Algorithms** - Bayesian method, Decision Trees, Random Trees, XGBoost, Support Vector Machine (SVM), KNN, Principal Component Analysis (PCA), Clustering, NLP.
- **Recommender Systems** - User-Based Filtering and Item-Based Filtering.
- **AutoML** - H2O, TPOT.
- **Platforms and Tools** - Numpy, Scipy, Tensorflow, Keras, HuggingFace, Pandas Profiling, Databricks, Sweetviz, Microsoft Tools(Excel, Outlook,Word, PowerPoint),pickle, JupyterLab, and graphviz.

DevOps

- ***D*ocker** - Basic understanding of creating custom Docker images using Dockerfile, running bash scripts inside docker container and managing images in docker hub.
- **Kubernetes** - Basic knowledge of orchestration tool for container management
- **Github** - Working knowledge of pull, commit, push and pull requests.
- **Cron jobs** - Writing cron expressions, Setting up recurring tasks, and monitoring crontab logs.

Miscellaneous skills

- **ChatGPT** - Prompts to generate ideas and codes.
- **Other Tools** - Mendeley, Microsoft Access, ChatDoc, Google Doc

---

PROJECTS

1. CHEST X-RAY CLASSIFICATION USING DEEP LEARNING FOR AUTOMATED DETECTION OF TUBERCULOSIS

Developed an automatic tuberculosis (TB) detection system using deep learning models, specifically convolutional neural networks (CNNs). Implemented data augmentation techniques and preprocessing steps to improve model performance. Worked on feature engineering to identify important attributes contributing to classification. Evaluated performance using various metrics like accuracy, sensitivity, specificity, precision, AUC, and F1 score. Achieved a 97.62% accuracy on the test set. Aimed to expedite TB diagnosis, reduce waiting periods, and assist in mass screenings for TB eradication. Additionally, Spirometry Interpreter is developed to identify the obstructive/restrictive pattern of respiratory diseases using FEV1/FVC values and COPD Exacerbation detector to predict the forthcoming exacerbations if present by prompting the patients to enter the severity of the listed symptoms represented in a form of Likert scale. The main webpage is structured in a way to address the emerging trends in pulmonary diseases, the ramifications due to climate change and the epidemiological trends of these CRDs.

Technology : HTML, CSS, Python, JavaScript, Kaggle image datasets, Keras, TensorFlow, Wamp Server, Google Collab, Visual Studio, pickle, and Pandas.

2. Analyzing the Pima Indian Diabetes dataset

Performed analysis on the Pima Indian Diabetes dataset to predict diabetes presence using medical predictor variables. Conducted data loading, exploration, and statistical summary. Employed feature selection techniques and classification models such as Decision Tree, Random Forests, SVC, XGBoost, Naive Bayes, and AutoSklearn to develop predictive models. Enhanced model performance by identifying and removing outliers using box plots. Subsequently, re-trained and fine-tuned models to optimize accuracy. Achieved notable results, with Random Forests and XGBoost achieving an accuracy of approximately 82%. These models highlight the potential of machine learning algorithms in predicting diabetes presence based on selected variables. The project encompassed crucial stages of data preprocessing, feature selection, model training, and evaluation. The findings offer valuable insights into the relationship between medical factors and diabetes prevalence in the Pima Indian population.

Technology : Scikit-learn, NumPy, Pandas, Seaborn, Matplotlib, and JupyterLab.

3. Amazon employer access prediction using H2O

Implemented an AutoML workflow using H2O AutoML to optimize access management processes by predicting employee access needs based on job roles at Amazon. Performed data preparation, exploration, and transformation into H2O format, ensuring data suitability for modeling. Trained and evaluated the H2OAutoML model, identifying a top-performing Stacked Ensemble model. Although the developed model did not attain a high ranking on the competition leaderboard, this project provided valuable insights into the potential of AutoML techniques for automating access prediction. Demonstrated expertise in implementing end-to-end AutoML workflows, leveraging advanced technologies, and employing data-driven decision-making strategies.

Technology : H2O AutoML, Pandas, Python, Kaggle dataset, and JupyterLab.

4. Tweet's scrapping app

Tweet's scrapping app is an application that allows users to extract tweets from scattered tweet data based on the search term,data range, and count of tweets to be scrapped given by the user. This app helps the user get specific tweet details of their interest. The app displays the scraped tweets in a dataframe for easy visualization. The number of tweets displayed is restricted based on the count of tweets specified by the user. The newest tweets are displayed first. Users can push the scraped tweets to the MongoDB database by clicking a button. Users can download the scraped tweets in CSV or JSON format. The Data retrieved can be used for sentiment analysis and trend analysis. This app was hosted on Hugging Face spaces.

Technology : python, Streamlit, pandas, Snscrape, pymongo, MongoDB atlas, Hugging face, Visual Studio, Jupyterlab.

5. Phonepe pulse visualization App

Phonepe pulse visualization app is an application that provides valuable insights and analysis of data from the PhonePe digital payment platform. The data was cloned from the GitHub repository and stored in the local system. The data in JSON format was transformed into CSV format. Extracting the data for different years and quarters from different categories like aggregate, map, and Top locations, along with data for India and states. A dataframe was created using each of these data. Further created Merged Dataframes of India and states for all categories, for easy retrieval of data from the dataframe. The resulting dataframes were used for creating Graphs, charts, and maps for visualization. Based on the different options chosen by the users for each analysis, visualizations were created accordingly to understand payment categories, transaction patterns, user activity, and top locations.The App is deployed by Streamlit Community Cloud by taking the code from the Github repository. This app can be used for market research, Geolocation analysis, tracking performance indicators, measuring success metrics, and identifying areas for improvement.

Technology - Python, Streamlit, Pandas, Plotly Express, Matplotlib, Geopandas, and Plotly Graph Objects, Visual Studio, Jupyterlab

6. YouTube Data Harvesting and Warehousing

YouTube Data Harvesting and Warehousing is an application that allows users to access and analyze data from multiple YouTube channels. Created a dashboard using the streamlit framework. Obtained API credentials by enabling the YouTube Data API in the Google Developers console. Using Google Api , extracted YouTube Channel, Playlist, Video and Comment data. The Main Page of the app contains information about the application and how to get the channel ID of the YouTube channel. Users should enter a Channel ID, and the data retrieved from YouTube will be stored in MongoDB. Afterwards, users can select the name of the channel from the dropdown menu to migrate the data to SQL. The user is supposed to click on the questions to view the results for the same in a Dataframe , which is possible by using SQL queries to retrieve the data from SQL database and displaying it in table/ table or dataframe using Pandas. This app was hosted on Hugging face spaces. Using this app, we can understand the performance, engagement, and trends of specific channels/specific genres over time and frame content strategies to increase channel popularity and performance.

Technology - Python, python-youtube, pandas, mysql-connector, PyMySQL, SQLAlchemy, Db4Free (A free platform to work with MySQL database), MongoDB, MySQL, Youtube API, Streamlit, Visual Studio, Jupyterlab.

7. AWS Lambda leveraged Server Monitoring and Slack Notification System

Designing an Automatic Data Collection and Storage System with AWS Lambda and Slack Integration for Server Availability Monitoring and Slack Notification using AWS Lambda, CloudWatch, Slack API.

Technology - Python, AWS Lambda, CloudWatch, Slack API, psycopg2, AWS SNS, Slack workspace

---

EDUCATION

Master Data Engineering
GUVI, Chennai

Bachelor of Science Honors
Biomedical Sciences
Sri Ramachandra Institute of Higher Education and Research, Chennai

Higher Secondary Education
Kumararani Meena Muthiah Matric Higher Secondary School, Chennai

Secondary Education
Kendriya Vidyalaya CLRI, Chennai

CERTIFICATION

Udemy

- Machine Learning, Data Science and Deep Learning with Python
- The Complete Data Structures and Algorithms Course in Python

LANGUAGES

- English - Full Professional Proficiency
- Hindi - Limited Working Proficiency
- Tamil - Native / Bilingual