PRANAV JAYANT MAHAJAN

Bryan, TX 77801 | (979) 422-7634 | pranav.m1124@gmail.com | linkedin.com/in/pranavmahajan11 | github.com/pmahajan11

Actively seeking Data Science Internships / Co-ops. Authorized to work in the USA under F1 Visa.

EDUCATION

Texas A&M University, College Station, TX | MS in Industrial Engineering | Specialization - Data Science *May 2023*

Relevant Coursework: Engineering Data Analysis, Statistics - Distribution Theory, Design of Experiments, Databases and Computational Tools for Big Data

Savitribai Phule Pune University, Pune, India | Bachelor of Engineering, Mechanical Engineering *May 2020*

SKILLS

- **Programming/Tools** - Python, R, Tableau, mySQL, PostgreSQL, Databases, Excel, FastAPI, AWS EC2, S3, GitHub, Git
- **Libraries** - NumPy, SciPy, Pandas, Dask, Matplotlib, Seaborn, Scikit-learn, NLTK, XGBoost, Keras, Streamlit, Flask, PySpark
- **Machine Learning** - Supervised Learning (Linear and Logistic Regression, k-NN, Naïve Bayes, SVM, Decision Trees, Ensemble Models), Unsupervised Learning (K-Means, Hierarchical Clustering, DBSCAN), Deep Learning (CNNs, RNNs)
- **Data Analysis** - Data wrangling, Exploratory Data Analysis, Statistical analysis, Hypothesis Testing, Time-Series Modeling, Data-driven Storytelling, Visualization, Model Deployment

WORK EXPERIENCE

Student Assistant - Machine Learning, Texas A&M University, College Station, TX. *January 2022 - Present*

- Employ machine learning techniques to predict myocardial properties of a human heart from Endo-static Pressure Volume Relationship (EDPVR) features.
- Oversee Finite Element Analysis simulations run on Texas A&M's supercomputing facility to generate the EDPVR input features.
- Examine, clean and process the simulation output data in Python using Pandas and NumPy and implement feature engineering.
- Develop a Deep Feedforward Neural Network model using TensorFlow and Keras, that minimizes the Mean Absolute Error and maximizes the R-squared, report the results to the supervising Professor and improve the model through an iterative process.
- Achieve a target R-squared value of 95% or greater while minimizing the prediction error.

PROJECT EXPERIENCE

Toxic Comment Classifier API *February 2022*

- Cleaned and preprocessed the text of over 150,000 comments and transformed the text into 300 dimensional vectors using pre-trained GloVe word vectors, balanced the data-set by upsampling the minority class.
- Trained Logistic Regression, Decision Tree, Random Forest, and XGBoost classifier models on the transformed data. Selected the Random Forest Classifier model based on Accuracy of 98.95%, Log Loss of 0.36, and AUC ROC of 0.9889 on the Test data.
- Built a Toxic (or Explicit) Comment Classifier REST API using the FastAPI framework with a backend Postgres database and user authentication feature.
- Deployd the API on an Ubuntu virtual machine on Cloud using DigitalOcean.

Microsoft Malware Classification *April 2021*

- Managed a real life malware data-set of size 200 GB on Google Cloud Platform.
- Conducted Exploratory Data Analysis on the data-set using NumPy, Pandas, Matplotlib and created a 2D visual representation by applying dimensionality reduction using TSNE to determine the distribution of malware classes in the data-set.
- Applied KNN, Logistic Regression, Random Forest Classifier and XGBoost Classifier models for classifying malware into 9 classes, and selected the best algorithm based on the value of Multiclass Log Loss metric.
- Fine-tuned the hyperparameters of the best Gradient Boosting model using Randomized Search Cross Validation and attained a Test Multiclass Log-Loss of 0.03.

Social Network Graph Link Prediction - Facebook Challenge *March 2021*

- Analyzed a directed graph data-set containing follower-followee node relationships of 1.7 million users.
- Performed Feature Engineering and created Similarity measure features (Jaccard and Cosine distance) and Graph features (Shortest path, Adar index, follow back, and Hits score) by leveraging NetworkX, Pandas and NumPy.
- Experimented with Random Forest and Gradient Boosting models to predict presence of a link between two nodes of the graph.
- Implemented a custom loop to find best model hyperparameters and achieved a Test F1-Score of 0.904 and Test AUC of 91%.