|  |  |  |
| --- | --- | --- |
| Avinash Tiwari  IoT - Team Member (Data Analyst)  +91 8669158497 | avinashsemcom0007@gmail.com |  LinkedIn: [Avinash\_Tiwari](https://www.linkedin.com/in/avinash--tiwari/) GitHub: [Avinash\_Tiwari](https://github.com/AVI18794)  Motivated data analyst with 3 years of experience. Passionate about building models that fix problems. Relevant skills include machine learning, deep learning, problem solving, programming, and logical thinking. | | |
|  |  |
| **Basic Skills:** Numpy, Matplotlib, Sci-kit Learn, Python,  OpenCV, Jupyter Notebook, TensorFlow, Anaconda,  SCIpy, Pandas, Seaborn, SQL, C++,Pytorch  Modin, Raspberry Pi, Streamlit, SpaCy, NLTK.  **Platforms:** IBM Watson Studio, Tableau, Alexa Skill Development,  Google Actions Console, SQL Server, Pycharm,  Visual Studio Code, Spyder, Linux, Google Colab,  GitHub. |  | **Education:**  2015-2018  MCA, D. Y. Patil Institute of MCA, Pune  | CGPA: 8.56  2012-2015  BCA, SEMCOM, Anand,Gujarat  | CGPA: 8.46 |
| Experience and Projects |  | Certifications |
|  |  |  |
| IoT - Team Member (Data Analyst): July 2018- Present  Kalyani Technologies Limited, Pune   * Developed knuckle part variant detection application using Deep Learning and OpenCV. * Developed Android Mobile application for real-time production KPI. * Successfully assessed the correlation of various parameters of the machine based on historical data. * Developed regression and anomaly detection model to forecast the parameters as part of the predictive maintenance project. * Developed QlikView dashboards for Finance * Energy consumption Analysis using Tableau and Python (Exploratory Data Analysis). * Developed utility programs for data gathering from multiple sources using Python (for data analysis preparation).   Deep Learning Intern: January 2018- July 2018  Techno Samarthyam, Ahmedabad   * Facial Recognition using deep learning. * OCR pipelining for text detection on receipt documents. * Exploration of various Neural Network Models for the object detection.   **Projects:**  Personal Projects  Data Analyst Nanodegree   * AB Testing model using Python and Statistical Techniques. * Data wrangling and Analysis using Python and SQL. * Chat analysis using python and Data Visualization. (For understanding what is the most discussed topics, most active time of users etc.)   Natural Language Processing Nanodegree   * PoS tagging including table lookups and n-grams using Python * Detecting deceptive review using NLP. * NER (Named Entity Recognition) using NLP. * Speech recognition by building end-to-end ASR by DNN. * Machine Translation from English to French by building a DNN. |  | * [Deep Learning Specialization](https://www.coursera.org/account/accomplishments/specialization/certificate/MMYXFAFK7L24) * [IBM Data Science Professional](https://www.coursera.org/account/accomplishments/professional-cert/certificate/L5W4ND4VV9JX) * [Fundamentals of Deep Learning for Computer vision (NVIDIA)](https://drive.google.com/file/d/1rPb4M-5Jm_dDZfGIk4zQtQXj_ptC6Xjl/view?usp=sharing) * [Data Scientist with Python-Datacamp](https://www.datacamp.com/statement-of-accomplishment/track/2c8769d4ec997412246d6fce4f00d37be4e95c10) * [Data Analyst Nanodegree by Udacity](https://confirm.udacity.com/FFNA2RRH) * [NLP Nanodegree by Udacity](https://confirm.udacity.com/FMPJJN2) |
| **Achievements:**   * Successful Deployment of the Alexa Skill Development application for Alexa. * Scholarship for Quantum Computing from Qubit by Qubit. * Quantum Machine Learning (QML) Scholarship by QWorld. * Scholarship for Data Analysis Nanodegree from Bertelsmann (Udacity). |
| Other Projects   * Object classification and detection for traffic signs classification using python and Deep Learning. * TV script Generation Using NLP. * Sentiment Analysis Using NLP. * Advance Lane Line Detection using OpenCV and Python. |
| * Facial Key point Detection using DLib and Python. * Automatic Image captioning using PyTorch and DL. * Document Redaction Using NLP. * Text Generation using Seq to Seq Model. * Contextual Chatbot Using Deep Learning (Pytorch) |