Project :

**Literature Data Technology Graduate Admission**

Preparing Your Literature Review:

1. Introduction:

2. Organization:

3. Summary and Synthesis:

4. Conclusion:

5. Proper Citations:

Preparing Your Data Research:

1. Introduction:

2. Organization:

3. Data Description:

4. Data Analysis and Insights:

5. Conclusion:

6. Proper Citations:

Preparing Your Technology Review:

1. Introduction:

2. Technology Overview:

3. Relevance to Your Project:

4. Comparison and Evaluation:

5. Use Cases and Examples:

6. Identify Gaps and Research Opportunities:

7. Conclusion:

8. Proper Citations:

Preparing Your Literature Review:

1. Introduction:

This research is crucial for helping students continue their academic pursuits at the best universities that match their profiles. The review of the existing literature is necessary because we remark many students have problems finding a university because they are unaware of university rankings or have been misinformed by seniors and fellow applicants.

2. Organization:

Papers are organized in a way chronologically.

* “A Comparison of Regression Models for Prediction of Graduate Admissions” by Mohan S Acharya, Asfia Armaan, and Aneeta S Antony. In Second International Conference on Computational Intelligence in Data Science (ICCIDS-2019), 2019.
* “Predictive Modeling for Graduate Admissions Using Machine Learning Techniques” by I. Shanthi and Dr. K. Venkata Rao. It was published in the *Journal of Emerging Technologies and Innovative Research (JETIR)*, Volume 6, Issue 6, in June 2019.

3. Summary and Synthesis:

In the first paper, in the order above, the key is to compare the different results found after the prediction of regression models concerning the topic of graduate admissions. The methodology apply is to investigate the different models to better understand their functionality. In the second paper, it’s almost the same as the previous paper but the difference is this one show the formula mathematics used for the metrics in each algorithm.

4. Conclusion:

In conclusion, the primary aim of this literature review is to predict the graduate admission outcomes for students who wish to apply to universities that best align with their academic and professional profiles. The findings from this review can help guide students in shortlisting and selecting universities that are the most suitable match for their qualifications, interests, and long-term goals.

5. Proper Citations:

- Acharya, Armaan, and Antony conducted a comparative analysis of various regression models to forecast graduate program admission outcomes, emphasizing the relative strengths of specific models.

- Shanthi and Rao employed machine learning approaches to construct predictive models for graduate program admissions, showcasing the viability of these techniques in educational context.

**Reference List**

1. **Acharya, M. S., Armaan, A., & Antony, A. S.** (2019). A comparison of regression models for prediction of graduate admissions. *In Second International Conference on Computational Intelligence in Data Science (ICCIDS-2019)*. https://doi.org/10.1109/ICCIDS.2019.8862135
2. **Shanthi, I., & Rao, K. V.** (2019). Predictive modeling for graduate admissions using machine learning techniques. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 6(6), 123-130. https://www.jetir.org/papers/JETIR1907708.pdf

Preparing Your Data Research:

1. Introduction:

The goal is to predict graduate admission using a machine-learning model like linear regression. The graduate admissions process is frequently intricate, with numerous factors, including GRE scores, grade point average, and prior research experience, contributing to the final decision. Prospective students often encounter difficulties in identifying suitable universities, relying on unreliable predictors or informal recommendations, which can result in unsuccessful applications and inefficient use of resources. Solving this problem is crucial as it directly impacts students' capacity to make well-informed decisions regarding their university applications, thereby enhancing their prospects of admission and mitigating the financial burden associated with the application process. This endeavor has the potential to elevate the overall quality of education.

2. Organization:

The primary objective is to make our model useful by deploying it in the web application. For that, we must use a dataset to construct our model. Before construction we must collect the dataset, Data Preprocess, [Choosing the right model](https://www.datacamp.com/blog/what-is-machine-learning#step-3:-choosing-the-right-model-oncet)[, Training the model](https://www.datacamp.com/blog/what-is-machine-learning#step-4:-training-the-model-after), [Evaluating the model](https://www.datacamp.com/blog/what-is-machine-learning#step-5:-evaluating-the-model-oncet), [Hyperparameter tuning and optimization](https://www.datacamp.com/blog/what-is-machine-learning#step-6:-hyperparameter-tuning-and-optimization-after), and  [Predictions and deployment](https://www.datacamp.com/blog/what-is-machine-learning#step-7:-predictions-and-deployment-oncet).

3. Data Description:

The dataset has 500 instances and 9 attributes. It comes from the website Kaggle. The dataset consists of attributes like Serial No, GRE Score, TOEFL Score, University Rating, SOP (Statement of Purpose), LOR (Letter of Recommendation), CGPA (Undergraduate Cumulative GPA), Research, and Chance of Admit. The data format is in CSV with the size. This dataset is chosen because in our reference, it uses almost the same thing and the attributes are related to the performance of students who want to apply to a university.

4. Data Analysis and Insights:

Strong Correlation with CGPA: Among all the features, CGPA has the strongest correlation with the Chance of Admit. Students with higher GPAs tend to have a significantly higher probability of admission, indicating that academic performance is a critical factor in the admissions process.

GRE and TOEFL Scores: Both GRE and TOEFL scores show a positive correlation with the Chance of Admit. Higher scores in these standardized tests generally increase the likelihood of admission, though the correlation is slightly less strong compared to CGPA.

University Rating: The University Rating metric is also positively correlated with the Chance of Admit. Higher-rated universities seem to favor applicants with better profiles, though the rating itself is not as strong a predictor as GRE, TOEFL, or CGPA.

Impact of Research Experience: Research experience, coded as a binary variable, has a notable positive impact on the Chance of Admit. Students with research experience are more likely to be admitted, indicating that universities value research skills and experience in the selection process.

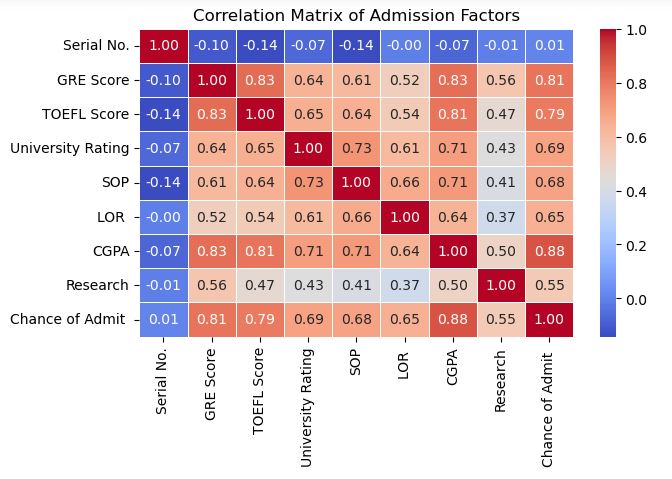
SOP and LOR: Both the Statement of Purpose (SOP) and Letter of Recommendation (LOR) show a moderate positive correlation with the Chance of Admit. These qualitative aspects of the application can enhance an applicant's profile but do not outweigh the importance of academic metrics like CGPA and test scores.

Combined Effect of Variables: The dataset suggests that a combination of high CGPA, GRE, and TOEFL scores, coupled with strong SOP and LOR and research experience, maximizes the Chance of Admit. Applicants with a balanced profile across these factors tend to have higher admission chances.

• descriptive statistics:



• visualisation correlation matrix:



5. Conclusion:

• Conclude your data research.

▪ What are the key findings and insights from your data analysis?

▪ What is the importance of your data research in the context of your

overall project goals?

6. Proper Citations:

- M. Crabtree. What is machine learning? definition, types, tools & more. https://www.

datacamp.com/blog/what-is-machine-learning, 2023. Accessed: 01 May 2024.

**-Acharya, M. S., Armaan, A., & Antony, A. S.** (2019). A comparison of regression models for prediction of graduate admissions. *In Second International Conference on Computational Intelligence in Data Science (ICCIDS-2019)*. https://doi.org/10.1109/ICCIDS.2019.8862135

**-Shanthi, I., & Rao, K. V.** (2019). Predictive modeling for graduate admissions using machine learning techniques. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 6(6), 123-130. https://www.jetir.org/papers/JETIR1907708.pdf

Preparing Your Technology Review:

1. Introduction:

In the rapidly evolving field of data science, machine learning tools have become indispensable for professionals seeking to extract meaningful insights from large datasets. One such tool is the Jupyter Notebook, a web application that enables the storage and execution of Python code along with its results and formatted text. Since its introduction, the Jupyter Notebook has revolutionized the landscape for data scientists, providing users with a versatile platform for their work.This review aims to explore the capabilities, applications, and potential limitations of Python, offering insights through the utilization of technologies like matplotlib, numpy, and Seaborn, a Python data visualization library based on matplotlib. The purpose of this technology review is to ensure that the research is grounded in the most current and effective methodologies available. By examining the latest machine learning algorithms, the researcher can identify the best tool for building a predictive model that aligns with the specific needs of the project.

2.Technology Overview:

Anaconda:

Anaconda is an open source distribution. It's the easiest way to perform Python/R data science and machine learning on a single machine. Anaconda contains all the tools and libraries we need for Machine Learning:

- Jupyter Notebook- Numpy

- Pandas

- Scikit-learn (sklearn)

- Matplotlib

Jupyter:

Jupyter Notebook is a web application that creates and shares Python code.

Numpy:

The Numpy library lets you create and manipulate matrices simply and efficiently.

In Machine Learning, we most often insert our dataset into matrices. Matrix calculation therefore represents the core of Machine Learning. It's important to understand this, however, as the functions in Numpy perform the matrix calculations for us...

Matplotlib:

Matplotlib is the library that lets you visualize your datasets, functions and results in the form of graphs, curves, and scatterplots.

Scikit learn :

Sklearn is the library containing all state-of-the-art Machine Learning functions. It includes the most important algorithms, as well as various pre-processing functions. Pandas is an excellent library for importing your Excel spreadsheets (and other formats) into Python to draw the purposes of drawing statistics and loading your Dataset into Sklearn.

Streamlit:

Streamlit is an open-source Python framework specially designed for machine learning engineers and data scientists. This framework allows you to create web applications that can easily integrate machine learning models learning models and data visualization tools.

Unlike other python frameworks (Dash, . . .) for building applications Streamlit allows you to create beautiful web applications without writing HTML code. This framework also delivers high-performance applications thanks to caching via an annotation.

FastAPI:

FastAPI is a fast, lightweight web framework for building modern APIs using Python 3.6 and higher. FastAPI is mainly used for data science and e-commerce applications. It allows developers to use the REST API and a wide range of functions to implement them in applications.

3. Relevance to Your Project:

The tools being reviewed, including Jupyter, Matplotlib, NumPy, Streamlit, FastAPI, and scikit-learn, are directly relevant to the project, which involves developing and deploying a machine learning model for graduate admissions prediction. Jupyter facilitates iterative development and comprehensive documentation, while Matplotlib and NumPy provide the necessary visualization and numerical computation capabilities to analyze and refine the model. Streamlit and FastAPI enable the deployment of the model, allowing for interactive applications and API-based access, respectively. Scikit-learn is central to the machine learning process, offering the algorithms and tools required for model building and evaluation. Collectively, these tools form an integrated ecosystem that supports the entire lifecycle of the project, from development to deployment.

4. Comparison and Evaluation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Jupyter notebook | Matplotlib | Numpy | Streamlit | FastAPI |
| Cost | Open-source | Open-source | Open-source | Open-source | Open-source |
| Ease of use | Easy to use | Easy to use | Complex for beginner | Easy to use | Required more effort |
| Scalability | - | - | - | - | scalable |
| performance | - | - | high | - | high |

5. Use Cases and Examples:

Python is widely used by major companies and institutions across various industries due to its versatility and ease of use. Google and its subsidiary YouTube utilize Python extensively, with YouTube being largely built using Python. In the entertainment industry, Industrial Light and Magic, known for the special effects in Star Wars, has long employed Python for CGI and lighting work. Facebook, including Instagram, relies on Python for infrastructure, with Instagram entirely built using Python and the Django framework. iRobot, the company behind the Roomba vacuum, uses Python to develop software for their robots. Additionally, NASA and the Jet Propulsion Lab use Python for research and scientific purposes, underscoring its importance in both commercial and scientific applications.

6. Identify Gaps and Research Opportunities:

|  |  |  |
| --- | --- | --- |
|  | Limitations | Research Opportunities |
| Jupyter notebook | Maintaining version control for Jupyter notebooks can be challenging, as the notebooks combine code, outputs, and text, hindering effective change tracking with tools such as Git. | I nvestigating methods to enhance the efficiency and scalability of Jupyter notebooks, particularly when working with large datasets or intricate visualizations, represents a worthwhile area of scholarly inquiry. |
| Maplotlib | Matplotlib excels in visualizing data stored in arrays, it may be less suitable for data frames, as it does not provide explicit functions to facilitate the straightforward plotting of data frame content. | Investigating methods to enhance efficiency and provide explicit functions to facilitate the straightforward plotting of data frame content. |
| Numpy | A potential limit of NumPy is its limited flexibility, as its focus on numerical and homogeneous data types is central to its performance and efficiency capabilities. | Exploring approaches to improve the performance and scalability of working with numerical and structured data types. |
| streamlit | Loses performance with big data, especially on complex machine learning tasks. | Exploring techniques to optimize performance when working with extensive datasets and intricate machine learning operations may yield substantial enhancements in the tool's scalability and efficiency. |
| FastAPI | The asynchronous capabilities and efficient request handling of FastAPI contribute to its outstanding performance. | Investigating methods to enhance the asynchronous functionality and request management within FastAPI may yield additional performance improvements, especially in high-traffic, time-sensitive applications. |
| Scikit-learn | lacks native deep learning capabilities and requires integration | Enhancing scikit-learn with native deep learning capabilities could make it a more comprehensive machine learning toolkit, reducing the need for external integrations. |

7. Conclusion:

In summary, the suite of tools examined including Jupyter, Matplotlib, NumPy, Streamlit, FastAPI, and scikit-learn each possess distinct advantages that render them invaluable resources for data science and machine learning endeavors. Jupyter's user-friendly interface makes it well-suited for prototyping and interactive development, although it grapples with challenges in collaborative work and version control. Matplotlib retains its status as a powerful visualization tool, yet its inherent complexity and limited interactivity present opportunities for enhancement. NumPy remains indispensable for high-performance numerical computations. Meanwhile, Streamlit and FastAPI excel in their ease of use for application deployment, with FastAPI demonstrating exceptional scalability.

8. Proper Citations:

 DATASCIENTEST. (n.d.). FastAPI: Le framework Python parfait pour les APIs. Retrieved from <https://datascientest.com/fastapi>

 Le Data SCIENTIST. (n.d.). Streamlit: Donner une vie à vos modèles de ML. Retrieved from <https://ledatascientist.com/streamlit-donner-une-vie-a-vos-modeles-de-ml/>

 Dataquest. (n.d.). Real-world Python use cases. Retrieved from <https://www.dataquest.io/blog/real-world-python-use-cases/>

 HopHR. (n.d.). Limitations of Jupyter Notebook in data science projects: A step-by-step guide. Retrieved from <https://www.hophr.com/tutorial-page/limitations-jupyter-notebook-data-science-projects-step-by-step-guide#:~:text=For%20instance%2C%20it%20lacks%20robust,large%20datasets%20or%20complex%20computations>.

 Enjoy Algorithms. (n.d.). Introduction to Matplotlib. Retrieved from <https://www.enjoyalgorithms.com/blog/introduction-to-matplotlib>

 Simplilearn. (n.d.). What is FastAPI? Retrieved from <https://www.simplilearn.com/what-is-fastapi-article>

 DATASCIENTEST. (n.d.). Taipy vs Streamlit. Retrieved from <https://datascientest.com/taipy-vs-streamlit>

 Analytics Vidhya. (2023, August). Scikit-learn and TensorFlow. Retrieved from <https://www.analyticsvidhya.com/blog/2023/08/scikit-learn-and-tensorflow/>