**Capstone Project Concept Note and Implementation Plan**

**Project Title:** Prediction of Graduate Admission

**Team Members**

1. Bamba GUEYE

**Concept Note**

**1. Project Overview**

Our capstone project, titled "Prediction of Graduate Admission," aims to help graduate students make informed decisions about university selection. Many face challenges in shortlisting universities due to limited knowledge of rankings and unreliable information from peers. This often leads to misjudgment of their academic profiles, wasting time and resources. Our project develops a comprehensive tool that predicts graduate admissions based on a student's academic profile. The tool provides personalized guidance to identify suitable universities, supporting individual educational goals and contributing to the broader Sustainable Development Goal of quality education. Our project aims to equip students with the necessary resources to navigate the intricate graduate admissions process, enabling them to identify and pursue higher education opportunities that align with their academic qualifications and aspirations. The potential impact of our solution extends beyond individual students. By optimizing the match between student profiles and university programs.

**2. Objectives**

The objective is to develop an application that assists graduate students in identifying universities that align well with their academic qualifications through a thorough evaluation, thereby preventing them from over- or underestimating their profile. To provide students with an evidence-based approach to selecting appropriate universities, thereby mitigating the apprehension and burden associated with the application process.

**3. Background**

Current solutions provide some assistance to students navigating the graduate admissions process, but they often lack personalization, accessibility, and accuracy. Our project takes a machine learning approach, which is necessary to address these shortcomings. By using data-driven predictions, we will empower students to make informed decisions, increasing their chances of being accepted to universities that match their academic profiles. This will contribute to a more equitable and efficient higher education system.

**4. Methodology**

For our "Prediction of Graduate Admission" project, we intend to leverage various machine learning techniques and methodologies to develop a robust predictive model. Specifically, we will explore four regression models: Linear Regression, Decision Tree Regression, Random Forest Regression, and Support Vector Regression.

To implement our project, we will utilize key frameworks and libraries, such as Python, scikit-learn, NumPy, Matplotlib, and Seaborn. These tools will enable us to effectively handle the data processing, model development, and evaluation aspects of our work.

In evaluating the performance of the different models, we will employ a range of evaluation metrics, including R-squared, Mean Squared Error, and Root Mean Squared Error. These metrics will provide us with a comprehensive understanding of the models' predictive capabilities and help us select the most suitable approach for our project.

Furthermore, we plan to leverage FastAPI for the deployment of our model and Streamlit to create an intuitive interface for the user form.

**5. Architecture Design Diagram**

Data collection (CSV file)

Data analyse

Data preprocessing

Test data

Model training and building

Model testing

Deployment API (FastAPI)

Interface (UI Streamlit)

**6. Data Sources**

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.

**7. Literature Review**

This literature review helps students in selecting universities aligned with their academic profiles, addressing the common issue of students being unaware of rankings or misinformed by peers. It starts with a comparison of regression models for predicting graduate admissions, followed by a study focusing on predictive modeling with detailed mathematical formulas. Both aim to understand the effectiveness of various models in predicting admission outcomes. The review concludes that accurately predicting admissions can guide students in choosing universities that match their qualifications and career goals, helping them make informed academic decisions.

**Implementation Plan**

**1. Technology Stack**

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.

Seaborn:

‘’Seaborn is a Python data visualization library based on [Matplotlib](https://matplotlib.org/). It provides a high-level interface for drawing attractive and informative statistical graphics’’.

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.

**2. Timeline**

• Provide a detailed timeline for the different stages of your project,

o Data collection and preprocessing

Additionally, it would be valuable to explore more advanced preprocessing techniques to enable deeper analysis of the data.

o Model development

I already developed the linear regression model it‘s remain other models.

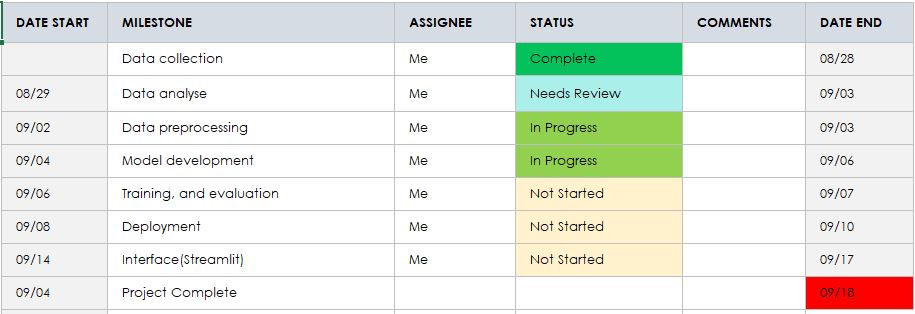
o Training, and evaluation

not yet

o Deployment

not yet

**3. Milestones**



**4. Challenges and Mitigations:**

o **Data quality:**

 Challenge**:** The quality of the data used in the project is crucial for building accurate and reliable predictive models. Potential issues include missing values, incorrect or inconsistent data, and insufficient data diversity.

 Strategy:

* **Data Cleaning and Preprocessing:** Implement rigorous data cleaning procedures to handle missing or incorrect values.

o **Model performance:**

 **Challenge:** Achieving high model performance can be difficult, especially when dealing with complex data or when the model overfits or underfits the training data.

 Strategy:

* **Cross-Validation:** Use techniques like k-fold cross-validation to evaluate model performance more robustly and prevent overfitting.
* **Hyperparameter Tuning:** Optimize model performance by tuning hyperparameters using grid search or random search methods.
* **Ensemble Methods:** Consider using ensemble methods like Random Forest or Gradient Boosting to improve predictive accuracy and reduce variance.
* **Model Evaluation:** Continuously monitor and evaluate model performance using various metrics (e.g., R-squared, Mean Squared Error) to ensure that the model meets the desired accuracy.

o **Technical constraints**.

 **Challenge:** Technical constraints such as computational resources, software limitations, or deployment challenges can hinder the development and deployment of the project.

 **Mitigation Strategy:**

* **Resource Management:** Optimize the use of available computational resources by employing efficient algorithms, reducing the complexity of the models, and using batch processing where possible.
* **Scalable Infrastructure:** Consider using cloud services (e.g., AWS, Google Cloud) that offer scalable resources to handle larger datasets and more complex models.
* **Tool Compatibility:** Ensure that the tools and libraries used in the project are compatible with the deployment environment. If necessary, consider containerization (e.g., Docker) to create consistent development and production environments.
* **Continuous Integration/Continuous Deployment (CI/CD):** Implement CI/CD pipelines to streamline the deployment process, reduce the risk of technical issues, and ensure that updates are smoothly integrated.

**5. Ethical Considerations**

Ensuring privacy begins with transparent and informed data collection practices, which involve obtaining informed consent from individuals before gathering their data. In our project, the dataset sourced from Kaggle is managed with the utmost privacy and security standards to preserve these principles. Furthermore, our examination of the dataset has not revealed any indications of selection bias, sampling bias, or other forms of bias that could adversely affect the target community. We are confident that we can obtain accurate evaluations by utilizing new data sources representative of the community.

**6. References**

 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/>

 Seaborn. (n.d.). Seaborn: Statistical data visualization. Retrieved from <https://seaborn.pydata.org/>

 OpenAI. (2024). ChatGPT response to project challenges and mitigation strategies. Retrieved August 25, 2024, from an interactive session with ChatGPT via OpenAI. <https://chatgpt.com/c/2226f60a-dae9-4f9a-a9e8-4e463192c02b>