**Architecture Design**

**CAMPUS PLACEMENT**

**PREDICTION**

**Index**

|  |
| --- |
| **Content** |
| Abstract |
| 1. Introduction |
| 1.1 What is Architecture Design? |
| 1.2 Scope |
| 1.3 Constraints |
| 2. Technical Specification |
| 2.1 Dataset |
| 2.2 Logging |
| 2.3 Database |
| 2.4 Deployment |
| 3. Technology Stack |
| 4. Proposed Solution |
| 5. Architecture Description |
| 6. User Input/Output Workflow |
|  |

**Abstract**

Machine Learning is a category of algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build models and employ algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. These models can be applied in different areas and trained to match the expectations of management so that accurate steps can be taken to achieve the organization’s target. In this paper, The Placement of students is one of the most important objective of an educational institution. Reputation and yearly admissions of an institution invariably depend on the placements it provides it students with. That is why all the institutions, arduously, strive to strengthen their placement department so as to improve their institution on a whole. Taking various aspects of a dataset collected for placement prediction, and the methodology followed for building a predictive model, results with high levels of accuracy are generated, and these observations can be employed to predict whether the student will be recruited in campus placements or not based on the available factors in the dataset.

**1. Introduction**

**1.1 What is Architecture Design?**

The goal of Architecture Design (AD) or a low-level design document is to give the internal design of the actual program code for the `Campus placement prediction`. AD describes the class diagrams with the methods and relation between classes and program specification. It describes the modules so that the programmer can directly code the program from the document.

**1.2 Scope**

Architecture Design(AD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. And the complete workflow.

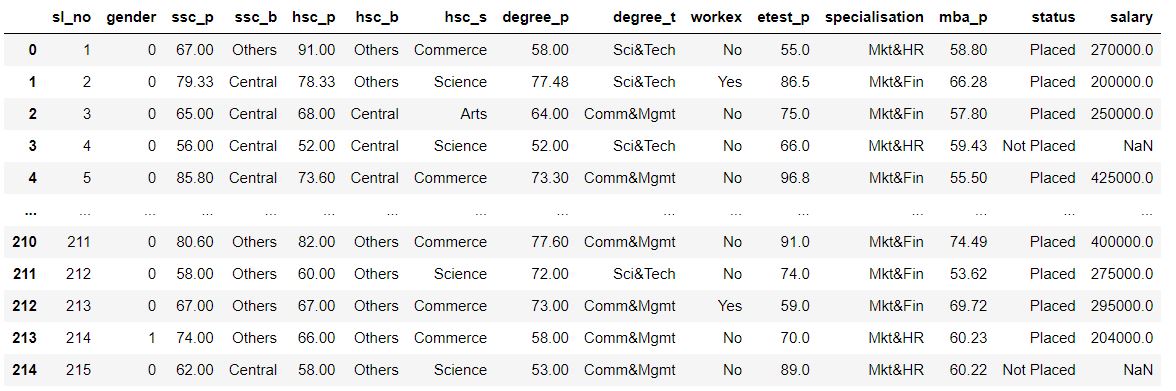
**1.3 Constraints**

We only predict the expected placement status based on the marks, subject and stream with imbalance data.

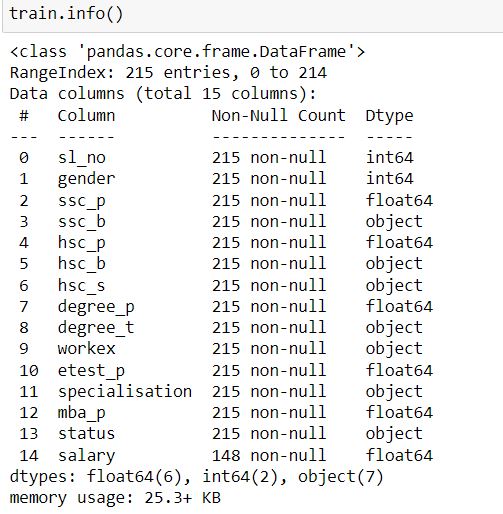
**2. Technical Specification**

**2.1 Dataset**

Data of students for predicting campus placement is collected from various institutions. This data about students contains their percentage at different educational level and which stream they choose in graudation and post graduation. The dataset looks like as follow:



The data set consists of various data types from integer to floating to object as shown in Fig.



In the raw data, there can be various types of underlying patterns which also gives an in-depth knowledge about the subject of interest and provides insights into the problem. But caution should be observed

with respect to data as it may contain null values, or redundant values, or various types of ambiguity, which also demands pre-processing of data. The dataset should therefore be explored as much as possible.

Various factors important by statistical means like mean, standard deviation, median, count of values and maximum value, etc. are shown below for numerical attributes.



Preprocessing of this dataset includes doing analysis on the independent variables like checking for null values in each column and then replacing or filling them with supported appropriate data types so that analysis and model fitting is not hindered from their way to accuracy. Shown above are some of the representations obtained by using Pandas tools which tell about variable count for numerical columns and model values for categorical columns. Maximum and minimum values in numerical columns, along with their percentile values for median, play an important factor in deciding which value to be chosen at priority for further exploration tasks and analysis. Data types of different columns are used further in label processing and a one-hot encoding scheme during the model building.

**2.2 Logging**

We should be able to log every activity done by the user

* The system identifies at which step logging require.
* The system should be able to log each and every system flow.
* Developers can choose logging methods. Also can choose database logging.
* The system should be not be hung even after using so much logging. Logging just because we can easily debug issuing so logging is mandatory to do.

**2.3 DataBase**

The system needs to store every request into the database and we need to store it in such a way that it is easy to retain and look into the records.

The system should capture every data that any user gave and the prediction that has been made by that input.

**2.4 Deployment**

For the hosting of the project, we will use AWS

**3. Technology Stack**

|  |  |
| --- | --- |
| Front End | HTML |
| Backend | Python |
| Deployment | AWS |

**4. Proposed Solution**

We will use performed EDA to find the important relation between different attributes and will use a machine-learning algorithm to predict Placement status. The client will be filled the required feature as input and will get results through the web application. The system will get features and it will be passed into the backend where the features will be validated and preprocessed and then it will be passed to a hyperparameter tuned machine learning model to predict the final outcome.

**5** **Architecture detail**



**5.1Data Gathering**

Data source: <https://www.kaggle.com/c/ml-with-python-course-project/data>

Train and Test data are stored in .csv format.

**5.2 Raw Data Validation**

After data is loaded, various types of validation are required before we proceed further with any operation. Validations like checking for zero standard deviation for all the columns, checking for complete missing values in any columns, etc. These are required because The attributes which contain these are of no use. It will not play role in predicting placement status.

Like if any attribute is having zero standard deviation, it means that’s all the values are the same, its mean is zero. This indicates that either the sale is increasing or decrease that attribute will remain the same. Similarly, if any attribute is having full missing values, then there is no use in taking that attribute into an account for operation. It’s unnecessary increasing the chances of dimensionality curse.

**5.3 Data Transformation**

Before sending the data into the database, data transformation is required so that data are converted into such form with which it can easily insert into the database. Here, if any column contains missing values is filled with proper value according to the data type. So they are filled in both the train set as well as the test set with supported appropriate data types.

**5.4 Data Preprocessing**

In data preprocessing all the processes required before sending the data for model building are performed. Here, I removed all those columns who standard deviation is zero because they are not contributing anything and those columns also which are of no use in predicting the status of Placement. After that I separate label and features columns from the train and test data.

**5.5 Feature Engineering**

After preprocessing we see that there are few categorical columns present in the data. For converting the categorical column, I have created the column transformer object and inside it I performed one hot encoding on categorical columns. Afterwars I save the transformer object so that it can be used for transforming the test and prediction data in same way as train data.

To handle imbalance data we have performed oversampling using imblearn smote function.

In the same step we also performed feature scaling to bring down every feature on the same page in terms of value range.

**5.6 Parameter Tuning**

I have used sklearn pipeline library with grid search cv to peform hyper parameter tuning. I have used different algorithms in pipeline and set different parameters for their important attributes.

**5.7 Model Building**

After doing all kinds of preprocessing operations mention above and performing scaling and hyperparameter tuning, the data set is passed into Logistic Regression,Decision Tree Classifier,Random Forest Classifier,SVC. It was found that SVC performs best with the highest accuracy score equals 0.88. So ‘Random forest classifier’ performed well in this problem.

**5.8 Model Saving**

Model is saved using pickle library.

**5.9 GitHub**

The whole project directory will be pushed into the GitHub repository.

**5.10 Deployment**

The cloud environment was set up and the project was deployed from GitHub into the Heroku cloud platform.

App link- <https://aditya-ineuron.herokuapp.com/>

**6. User Input / Output Workflow.**

****