Application Management Using Machine Learning:

About Dataset:

The datasets consist of several Application predictor variables. Predictor variables includes Application_id, Degree, University Year, PG and UG scores and so on. This dataset Consists of 14 attributes and 611 instances.

The attributes in the dataset are as follows:

Application_ID - ID for their application

Current City - City of the candidate

Python (out of 3) - Their Knowledge is Python one the Scale of 3

R Programming (out of 3) - Their Knowledge is R one the Scale of 3

Data Science (out of 3) - Their Knowledge is Data Science one the Scale of 3

Other skills - Additional Skills of the candidate

Institute - Institution of the Candidate

Degree - Degree of the Candidate

Stream - - Degree of the Candidate

Current Year of Graduation - Graduation Area of the Candidate

Performance_PG - PG Score of the candidate

Performance_UG - UG Score of the candidate

Performance_12 -12th Score of the candidate

Performance_10 - 10th Score of the candidate

Goal:

To build a Machine Learning model that helps the Internet buddy management to shortlist the potential candidates during their application process.

To Build a Machine Learning Model we need the target variable. For building the target variable we are having set of criteria. Using those criteria, we need to build the target variable.

Loading Packages:

First, we are loading required packages for performing our descriptive analysis and Building Machine Learning models for classification of the application.

Target or Class Variable :

For building the target Variable, we are using some conditional statements for using those criteria. Here we are adding new column weightage, which contains the score of the particular candidate. Here the score is calculate using those criteria.

Then we are building the class variable, based on the weightage. If the weightage is greater than 40, we are assigning 1 to class variable. If it is less than 40, we are assigning 0 to class variable.

Now we have our Class variable. So, we can train out model, for training our model we need to preprocess the dataset.

Skills count :

In our Dataset, we have a column called Other skills. This column contains various skills, but we need to assign scores for only certain skills. So, we are taking the skill count for each candidate. The skill count consists of count of the skills that they have and mentioned in criteria, i.e., if they have some skill and it is also present in the criteria, then the count is increased.

Dropping out the columns:

Before training our model, we need to take the variables that have effect on our class variables, i.e., we need to drop the unwanted columns from our dataset.

Degree:

In our Dataset, we have column called Degree. That columns also contain string data, so we are label encoding the columns and converting that into numerical ones.

Label encoding:

It refers to converting the labels into numeric form, so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

By performing all these steps, our dataset is ready for training process.

Choosing appropriate model:

Our goal is to find, if the candidate's application is selected or not. So here the we have only binary solution either yes or no, so we can use Logistic regression.

Logistic Regression:

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. It is the go-to method for binary classification problems (problems with two class values). Logistic Regression is used when the dependent variable(target) is categorical.

Using Logistic Regression, we are training and testing our model.

Now we need to deploy this model in cloud. This is done with the help of Flask.

For deploying in cloud, first we need to dump our project using joblib.

Joblib:

Joblib is a set of tools to provide **lightweight pipelining in Python**. In particular:

1. transparent disk-caching of functions and lazy re-evaluation (memorize pattern)
2. easy simple parallel computing

Once the code is dumped, we need to use that in flask.

Flask Python:

It is a web app framework, meaning it is a third-party Python library used for developing web applications. flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine.

HTML:

Using HTML, we need to create the form using given pattern. It is used for creating our own webpage.

By performing all these steps, the model will be deployed in the cloud.

Link of the application management web page:

http://application-process.herokuapp.com/