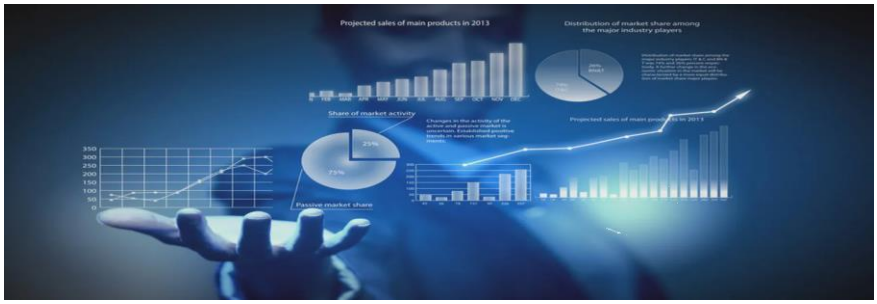
HR Analytics: Job Change of Data Scientists



Design:

The study aims to find out what are affective factors on candidate decision, predict the probability of a candidate to look for a new job.

Data

This dataset can be found at Kaggle in this link: HR Analytics: Job Change of Data Scientists | Kaggle

This dataset includes 19158 candidates with 14 attributes :

Column name	Column description
enrollee_id	Unique ID for enrollee
city	City code
city_development_index	Developement index of the city (scaled)
gender	Gender of enrolee
relevent_experience	Relevant experience of enrolee
enrolled_university	Type of University course enrolled if any
education_level	Education level of enrolee
major_discipline	Education major discipline of enrolee
experience	Enrolee total experience in years
company_size	No of employees in current employer's company
company_type	Type of current employer
last_new_job	Difference in years between previous job and current job
training_hours	training hours completed
target	0 – Not looking for job change. 1 – Looking for a job change

The dataset is available as the .csv file. a sample of data is shown in the following table:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_si
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20	Na
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	15	50-
2	11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	5	Na
3	33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree	<1	Na
4	666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>20	50-

Algorithms:

First, I downloaded the data from Kaggle website and read it.

Then, I started to clean the dataset by:

- Changed the type of column (target) to integer.
- Handle Missing Values:
  Imputing missing values in the data with mode.
- Drop city and enrollee id.
- Converts all values of all feature into numbers (category).

After that, I explored the data and showed the data to find the relation between the features.

Then I use SMOTE to oversample the imbalanced in Data ,To see the performance of these models I split my data into 80%(train- validation) sets /20% test set, and fit the models on train set, and test it on the validation and test sets, after that Scaling the data using standardization and implemented a multiple models to predict target:

- LogisticRegression
- RandomForestClassifier
- KNeighborsClassifier
- DecisionTreeClassifier
- Support Vector Classification

The best Model to predict the target was the RandomForestClassifier

Model	precision	recall	f1-score
Random Forest Classifier*	0.80	0.81	0.81
Decision Tree Classifier	0.77	0.77	0.77
Random Forest Classifier	0.73	0.73	0.73
Support Vector Classification	0.71	0.58	0.64
Logistic Regression	0.70	0.59	0.64
KNeighborsClassifier	0.66	0.66	0.66

This are the scores of all the models I did:

Testing Results:

	precision	recall	f1-score	support
0	0.81	0.80	0.81	2877
1	0.80	0.81	0.81	2876
accuracy			0.81	5753
macro avg	0.81	0.81	0.81	5753
weighted avg	0.81	0.81	0.81	5753

RandomForestClassifier*

* before I do "feature importance"

Testing Results:

	precision	recall	f1-score	support
0	0.64	0.75	0.69	2877
1	0.70	0.59	0.64	2876
accuracy			0.67	5753
macro avg	0.67	0.67	0.66	5753
weighted avg	0.67	0.67	0.66	5753

LogisticRegression

Testing Results:

	precision	recall	f1-score	support
0	0.73	0.74	0.73	2877
1	0.73	0.73	0.73	2876
accuracy			0.73	5753
macro avg	0.73	0.73	0.73	5753
weighted avg	0.73	0.73	0.73	5753

RandomForestClassifier

Testing Results:

	precision	recall	f1-score	support
0	0.77	0.77	0.77	2877
1	0.77	0.77	0.77	2876
accuracy			0.77	5753
macro avg	0.77	0.77	0.77	5753
weighted avg	0.77	0.77	0.77	5753

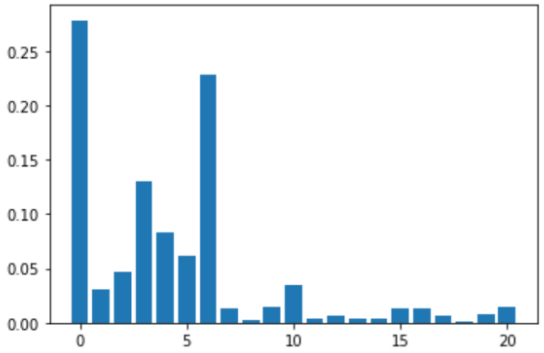
Decision Tree Classifier

Testing Results:

	precision	recall	f1-score	support
0	0.65	0.76	0.70	2877
1	0.71	0.58	0.64	2876
accuracy			0.67	5753
macro avg	0.68	0.67	0.67	5753
weighted avg	0.68	0.67	0.67	5753

Support Vector Classification

Feature importance:



Based on the bar plot above we drop all feature except city_development_index, experience_Imputed and training_hours.

Tools:

There are tools that will be used to achieve the goal of this study, such as: numpy, pandas, matplotlib , sklearn and seaborn for discovering and showing the data and train a model. The work will be done through Jupyter notebook using python.