Shreya Hunur
015269501
March 20, 2022

CMPE 257 Lab 1 Report

Dataset : Job Change Indicators

Observations:

1. Null values:

index	0
city	0
city_development_index	0
gender	3393
relevent_experience	0
enrolled_university	292
education_level	338
major_discipline	2089
experience	45
company_size	4430
company_type	4598
last_new_job	327
training_hours	0

There were many null values in the dataset. I have tried various methods to deal with the null values.

Methods used:

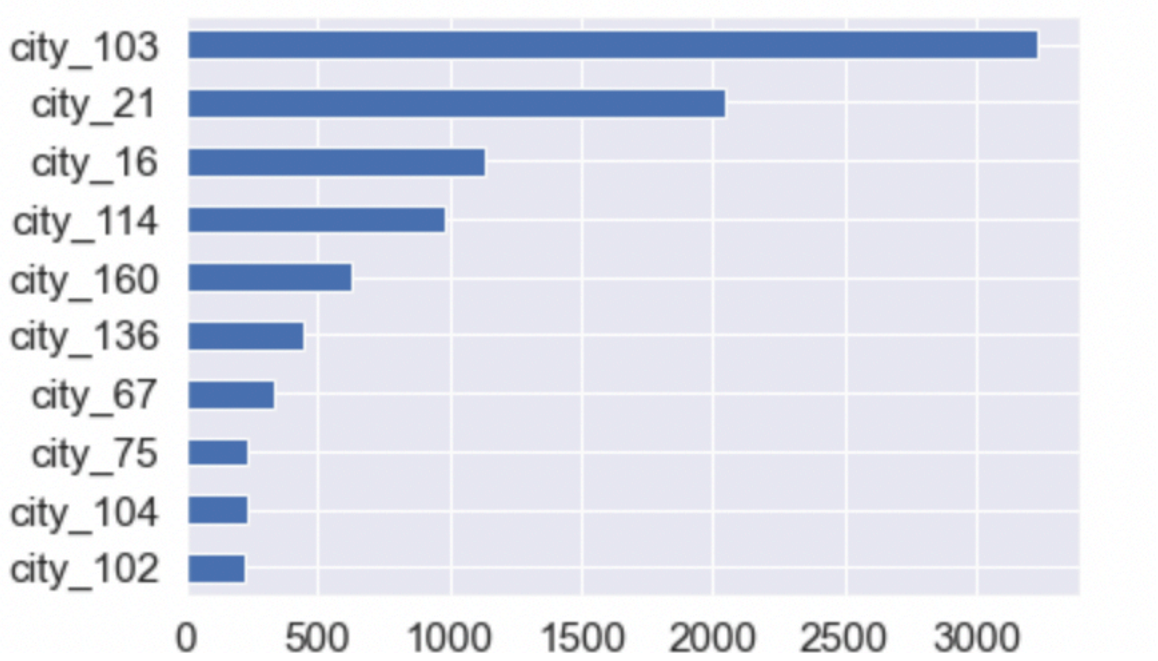
1. Impute these null values with mode.
2. Using KNN Imputer to impute these values.
3. Drop records which have null values.

The best solution was to impute the null values of mode.

2. Categorical features

After analyzing the features we see that most of them are categorical.
Since most of the features were categorical, I used Label encoding to transform the features.
I also used another method where I defined functions to manually replace non- numerical values.

3. City counts

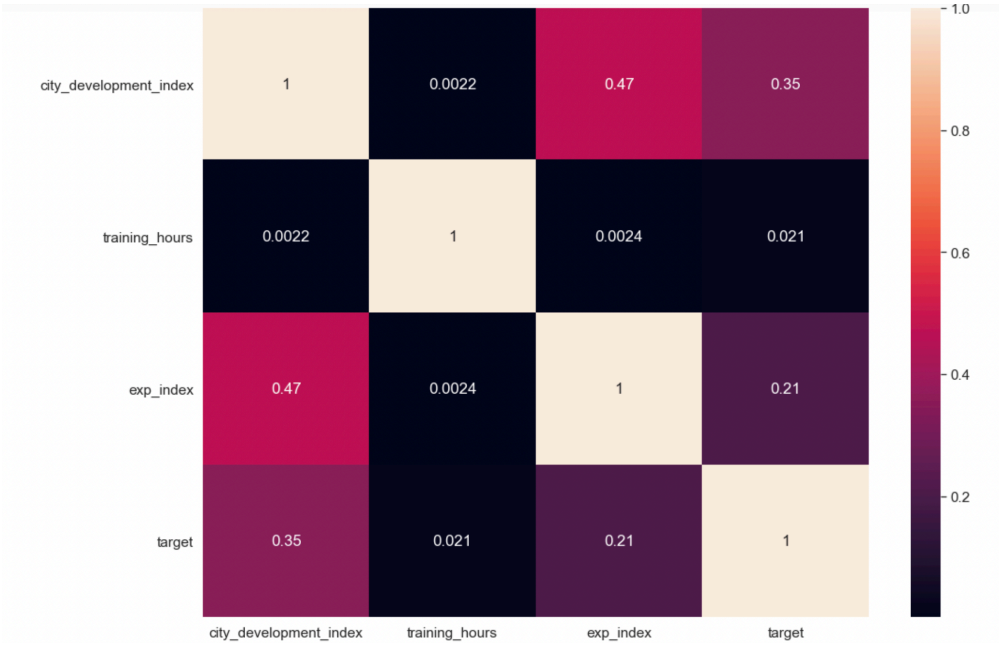


Value counts show that city 103 has most number of people. Our data has more number of people from city 103.

4. Feature Engineering
Created a new variable which combines city_development_index and experience to get experience index i.e. exp_index.

This helps a lot since it creates a new feature which has high impact on the target variable and helps in predicting better.

5. Tests to determine co-relation between variables
Performed Pearson's correlation, T test and Chi Square test to find out that only two features have high correlation with the target variable. And two variables are highly co-related with each other.



As we can see the exp_index has high correlation with target !

4. Train models
I have trained multiple models to train and test the data.
Using the different models classifiers since we are predicting whether the person will change the job or not.

Models I have trained include K-nearest neighbors, Logistic Regression, Decision Tree Classifier, random forest classifier and Neural Networks. I used GridSearch CV and hyper parameter tuning to fit the model better.

The training data given was split into train and test for each of the above models. Since the data was transformed well, all models definitely gave decent results.

However the random forest classifier performed exceptionally well with an 83 % accuracy.

5. Testing the data in Kaggle competition

I used the random forest classifier since it did best on training data with an accuracy of 83 percent. On Kaggle, it gave an accuracy of 0.5.