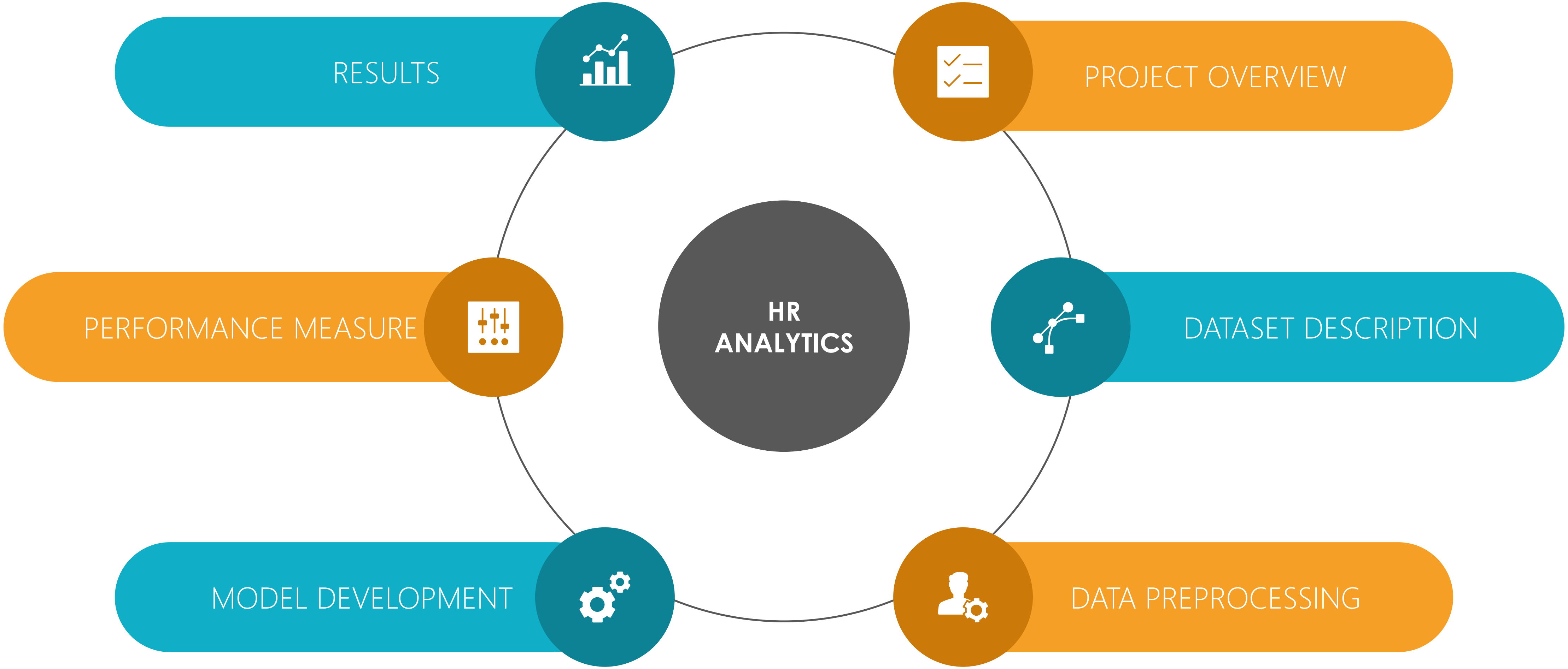
HR Analytics : Job change

Presented by
Adina Dingankar
Pranay Bhakthula
Rehapriadarsini Manikandasamy

Submitted for
DATS 6103

Presentation Outline



HR ANALYTICS

PROJECT OVERVIEW

DATASET DESCRIPTION

DATA PREPROCESSING

MODEL DEVELOPMENT

PERFORMANCE MEASURE

RESULTS

Project Overview

- The focus of this project is to predict the probability of a candidate to look for a new job or who will continue to work for the company

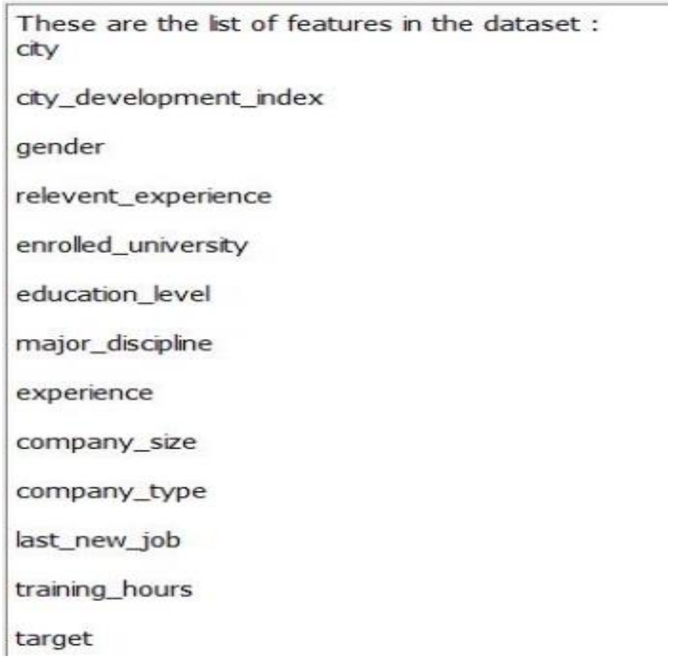
- It will be demonstrated using three machine learning algorithms:
    - Decision Tree Classifier
    - Random Forest Classifier
    - Support Vector Classifier

- Developed a GUI based application to display the end-to-end modelling

Dataset Description

- The dataset used has educational and professional records of various candidates who have completed training in a company
- The dataset has 19158 observations and 14 features, most of them are categorical

  Source: https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists
- 8 amongst 14 features have missing values



These are the list of features in the dataset :
city

city_development_index

gender

relevent_experience

enrolled_university

education_level

major_discipline

experience

company_size

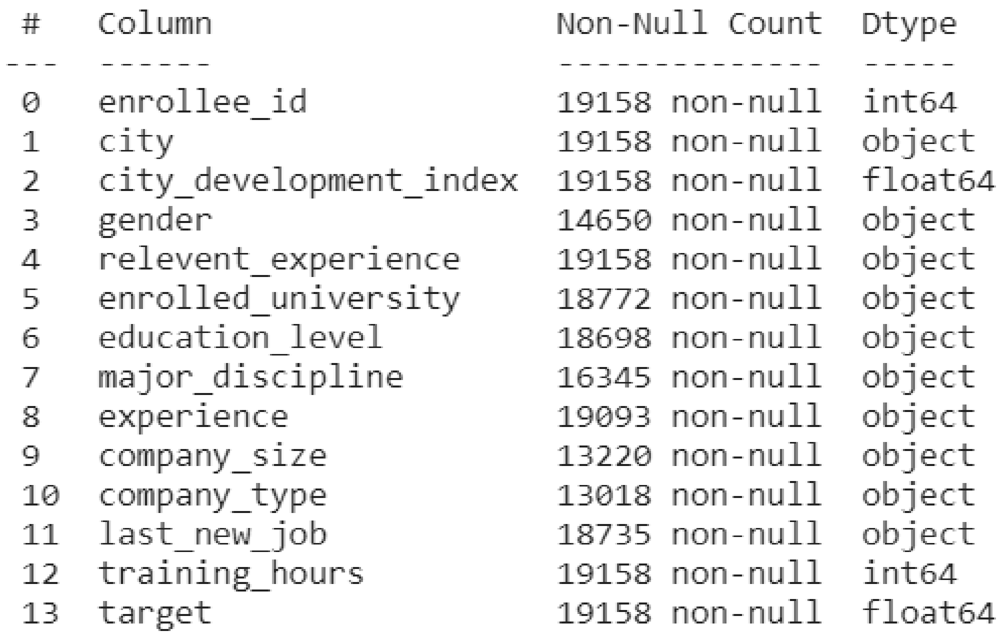
company_type

last_new_job

training_hours

target

Data Preprocessing

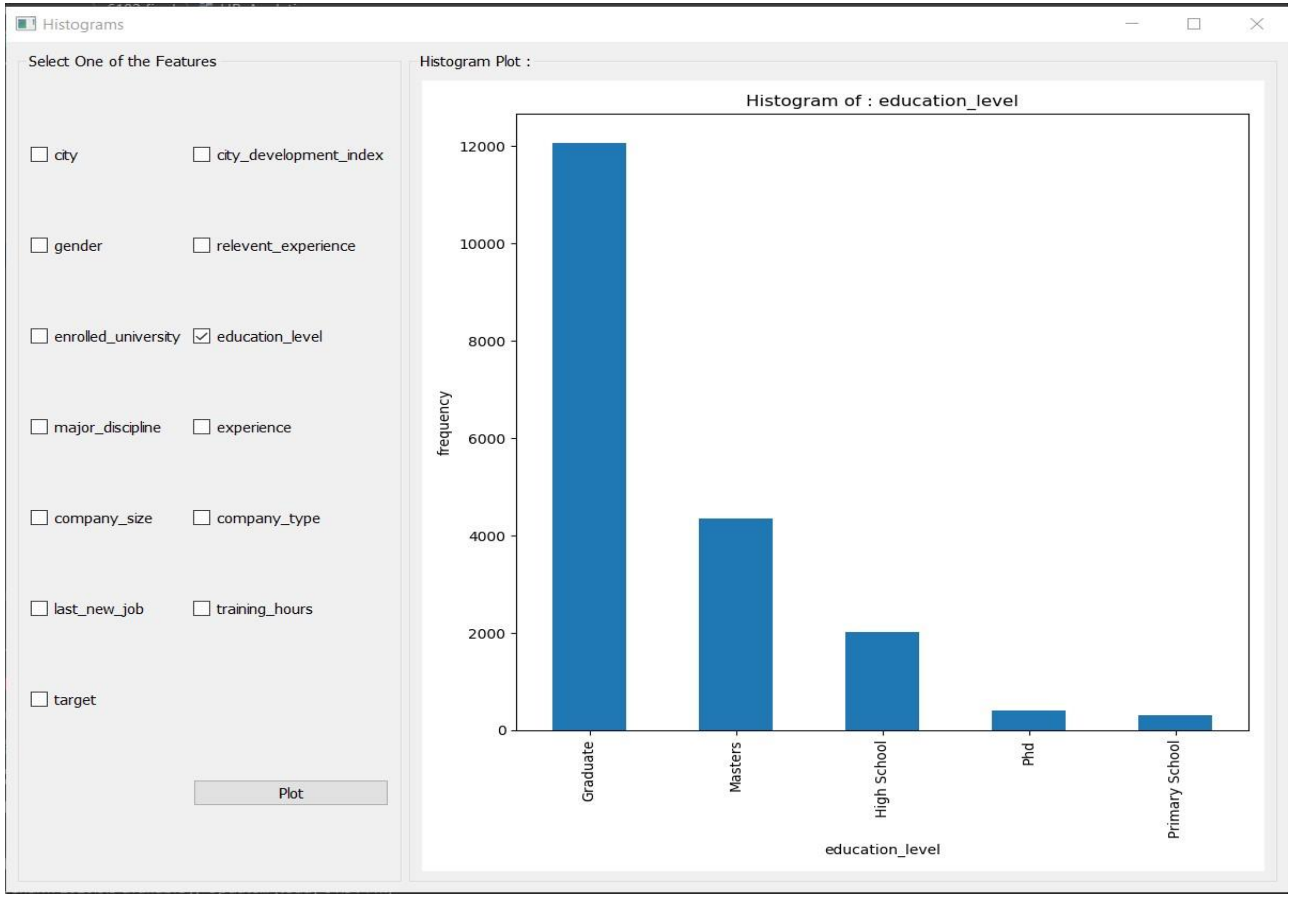


```
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   enrollee_id             19158 non-null  int64
 1   city                    19158 non-null  object
 2   city_development_index  19158 non-null  float64
 3   gender                  14650 non-null  object
 4   relevent_experience     19158 non-null  object
 5   enrolled_university     18772 non-null  object
 6   education_level         18698 non-null  object
 7   major_discipline        16345 non-null  object
 8   experience              19093 non-null  object
 9   company_size            13220 non-null  object
 10  company_type            13018 non-null  object
 11  last_new_job            18735 non-null  object
 12  training_hours          19158 non-null  int64
 13  target                  19158 non-null  float64
```

- Features with null values are updated with maximum value count of their respective columns
- The column enrollee_id is dropped , since it doesn't have much influence on target
- Label Encoder is applied to the features as our use case being the classification problem
- Encoding is done to decide in a better way on how these labels must be operated and labels are converted into numeric form

EDA Analysis

- EDA analysis option allows user to visualize histograms and scatter plots of variables



Histograms

Select One of the Features

☐ city ☐ city_development_index

☐ gender ☐ relevent_experience

☐ enrolled_university ☑ education_level

☐ major_discipline ☐ experience

☐ company_size ☐ company_type

☐ last_new_job ☐ training_hours

☐ target

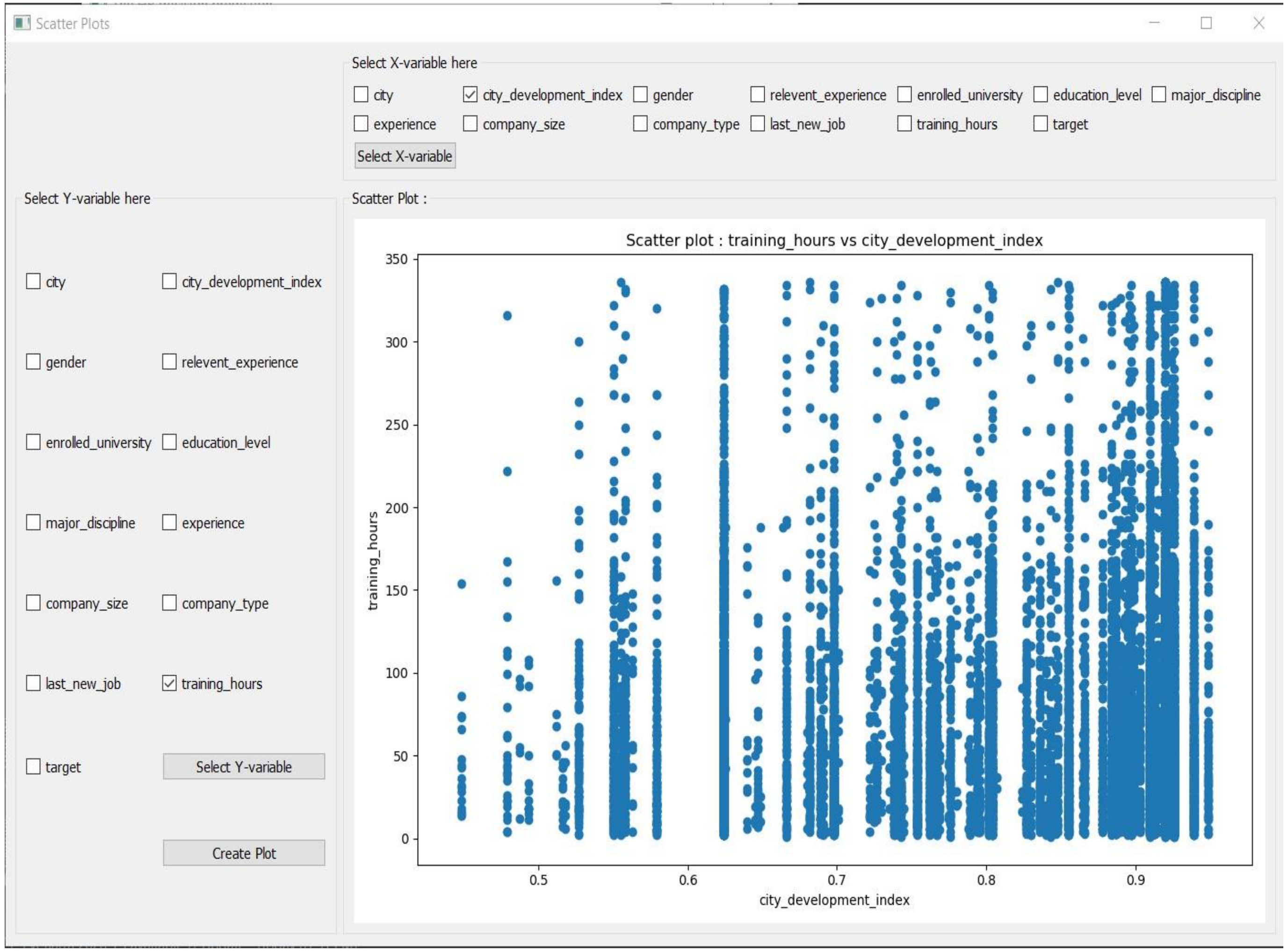
Plot

Histogram Plot :

- This histogram sample provides the distribution of employee's based on their education level
- Likewise, the user can view the distribution of other variables

EDA Analysis

Scatter Plot



Scatter Plots

Select X-variable here

□ city ☑ city_development_index □ gender □ relevent_experience □ enrolled_university □ education_level □ major_discipline
□ experience □ company_size □ company_type □ last_new_job □ training_hours □ target

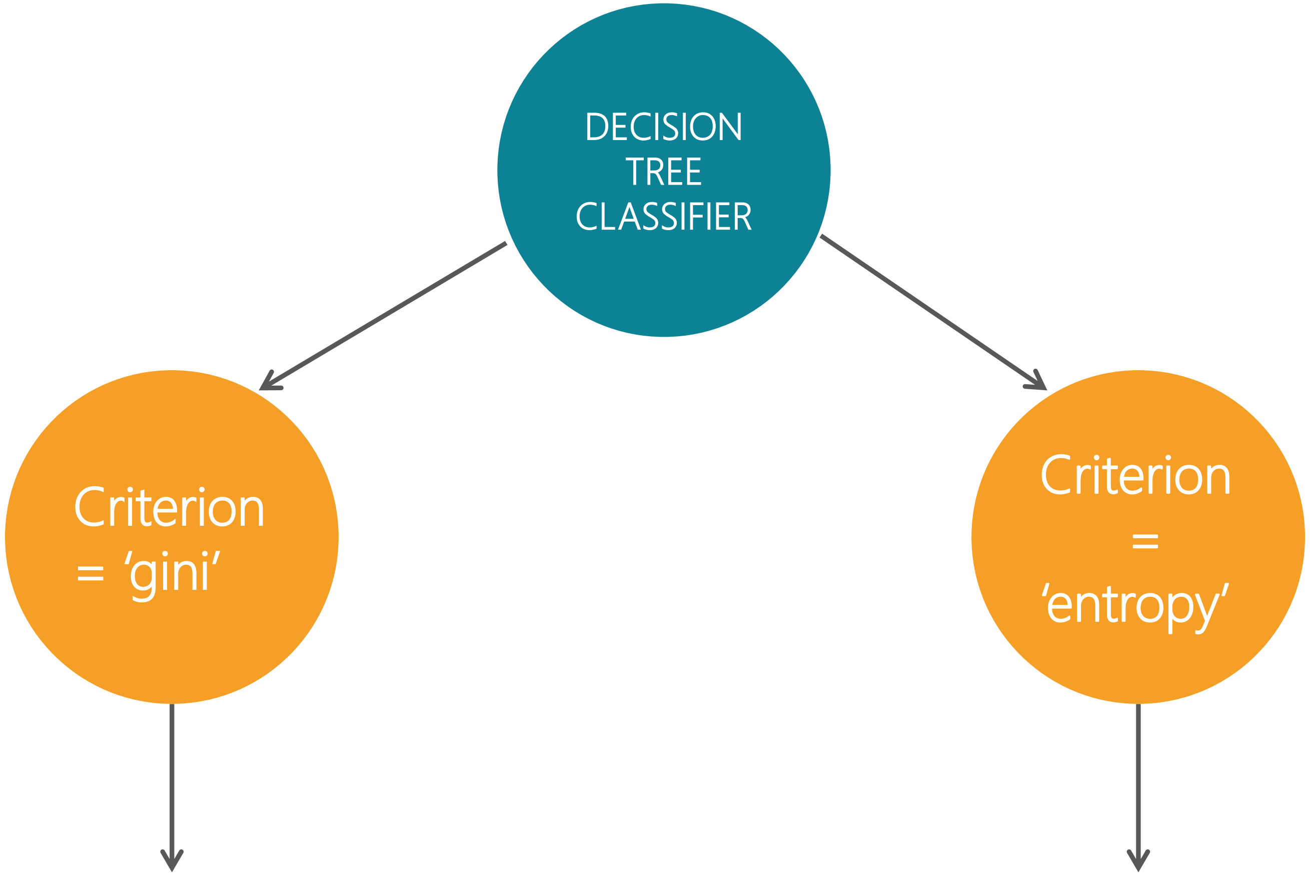
Select Y-variable here

□ city □ city_development_index
□ gender □ relevent_experience
□ enrolled_university □ education_level
□ major_discipline □ experience
□ company_size □ company_type
□ last_new_job ☑ training_hours
□ target

Scatter Plot :

The scatterplot sample visualizes the training hours of employee against the city development index

Model Development



DECISION TREE CLASSIFIER

Criterion = 'gini'

Criterion = 'entropy'

```
self.clf_df_gini =
DecisionTreeClassifier(criterion="gini",
random_state=100, max_depth=vmax_depth,
min_samples_leaf=5)
```

```
self.clf_df_entropy =
DecisionTreeClassifier(criterion="entropy",
random_state=100, max_depth=vmax_depth,
min_samples_leaf=5)
```

Model Development

RANDOM FOREST CLASSIFIER

- The dashboard is populated using the parameters chosen by user
- The parameters are processed to execute in Sci-Kit learn Random Forest algorithm

Model 1

```
self.clf_rf_gini =
RandomForestClassifier(n_estimators=
n_esti, criterion='gini',
random_state=100)
```

Model 2

```
self.clf_rf_entropy =
RandomForestClassifier(n_estimators=
n_esti, criterion='entropy',
random_state=100)
```

Model Development

SUPPORT VECTOR CLASSIFIER

- The kernel preference and test size can be provided by the user
- SVC model constructed uses radial basis function kernel in default

Model

```
self.clf_svc = SVC(kernel=kernel1)

self.clf_svc.fit(X_train, y_train)

y_pred = self.clf_svc.predict(X_test)

y_pred_score = self.clf_svc. decision_function(X_test)
```

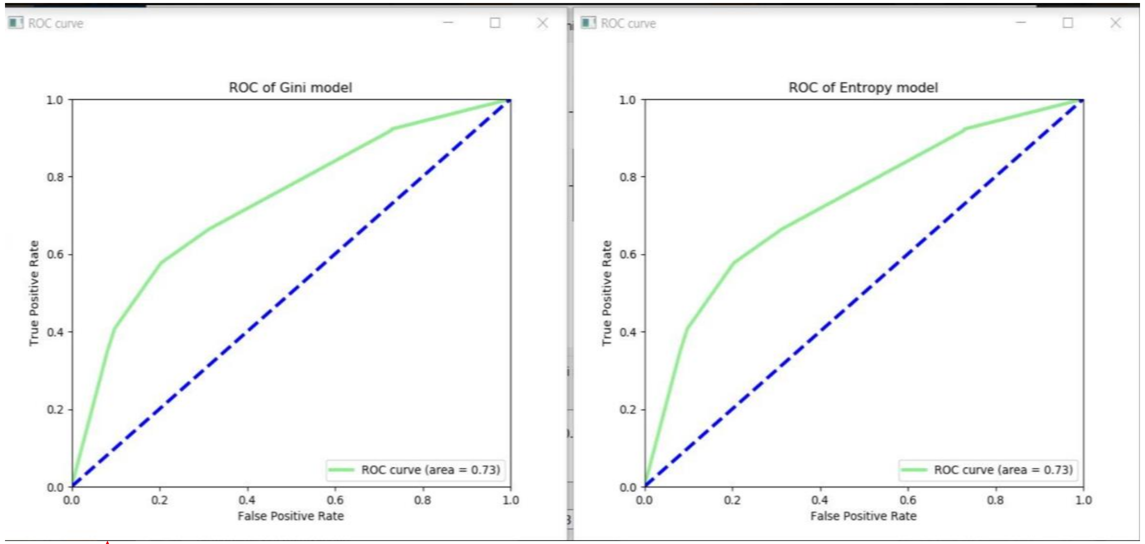
Performance Measure

The performance of the models are measured by :

- Confusion matrix
- Classification report
- Accuracy score
- Roc_auc curve

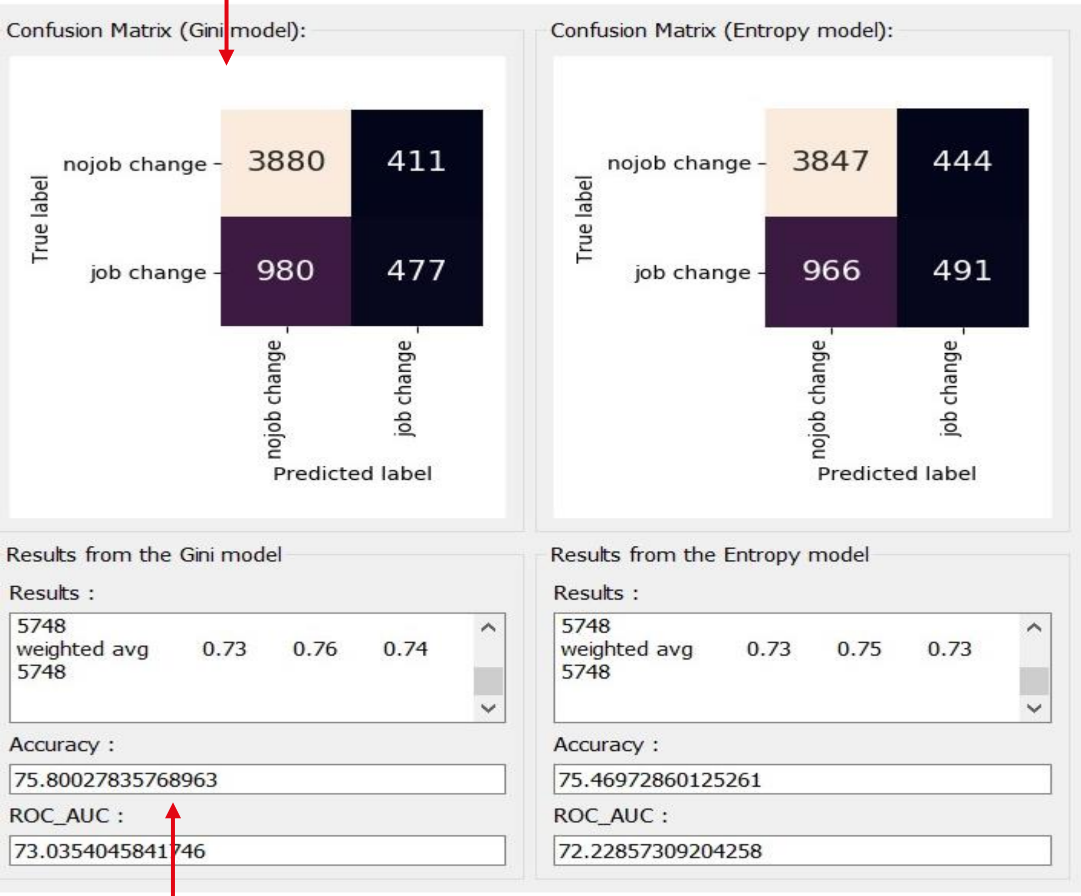
Performance Measure

The number of correct and incorrect predictions are summarized with count values and broken down by each class



ROC is a probability curve and AUC represents the degree or measure of separability

The report shows the main classification metrics precision, recall and f1-score on a per-class basis



Confusion Matrix (Gini model):

True label	nojob change	job change
nojob change	3880	411
job change	980	477

Predicted label

Confusion Matrix (Entropy model):

True label	nojob change	job change
nojob change	3847	444
job change	966	491

Predicted label

Results from the Gini model

Results :

```
5748
weighted avg       0.73      0.76      0.74
5748
```

Accuracy :

75.80027835768963

ROC_AUC :

73.0354045841746

Results from the Entropy model

Results :

```
5748
weighted avg       0.73      0.75      0.73
5748
```

Accuracy :

75.46972860125261

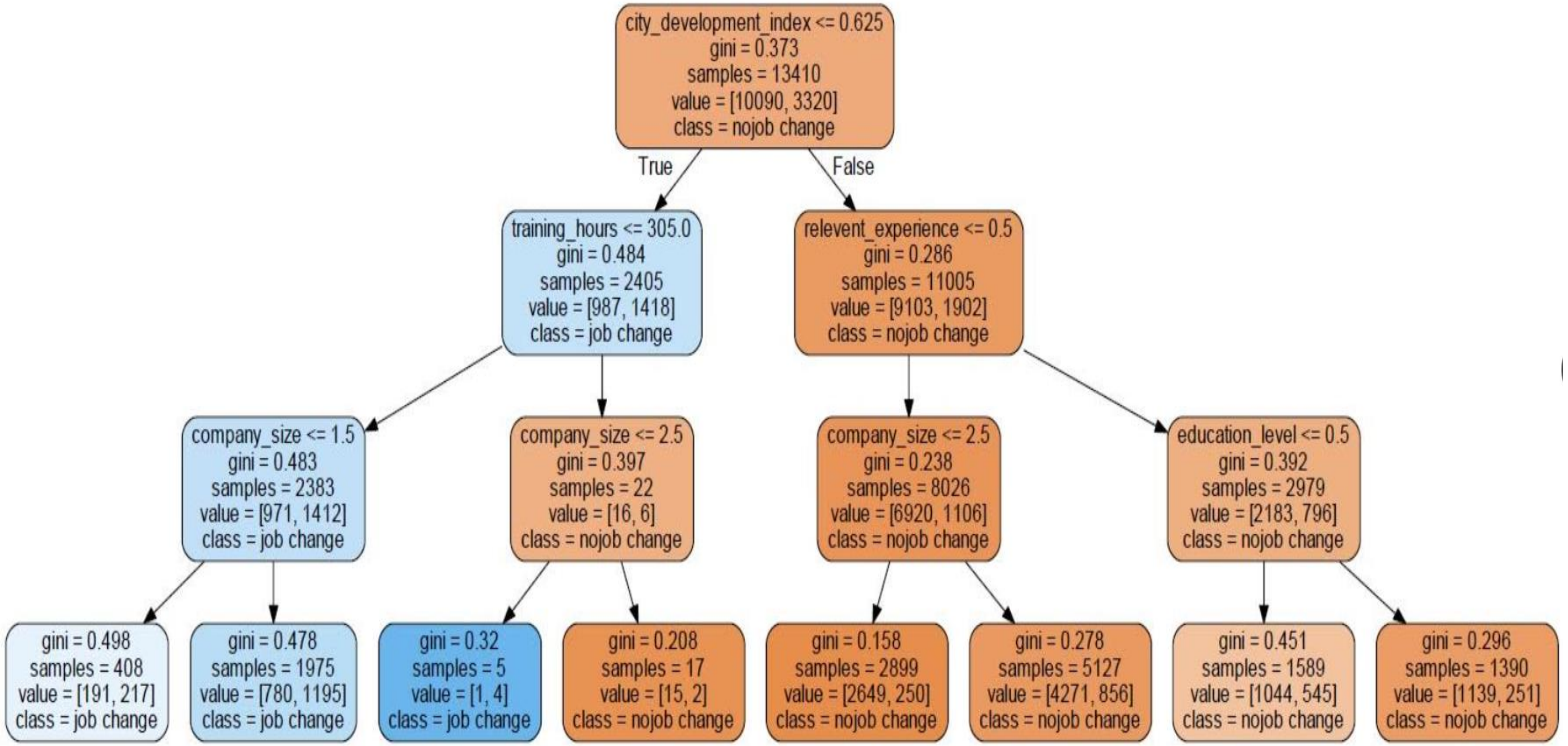
ROC_AUC :

72.22857309204258

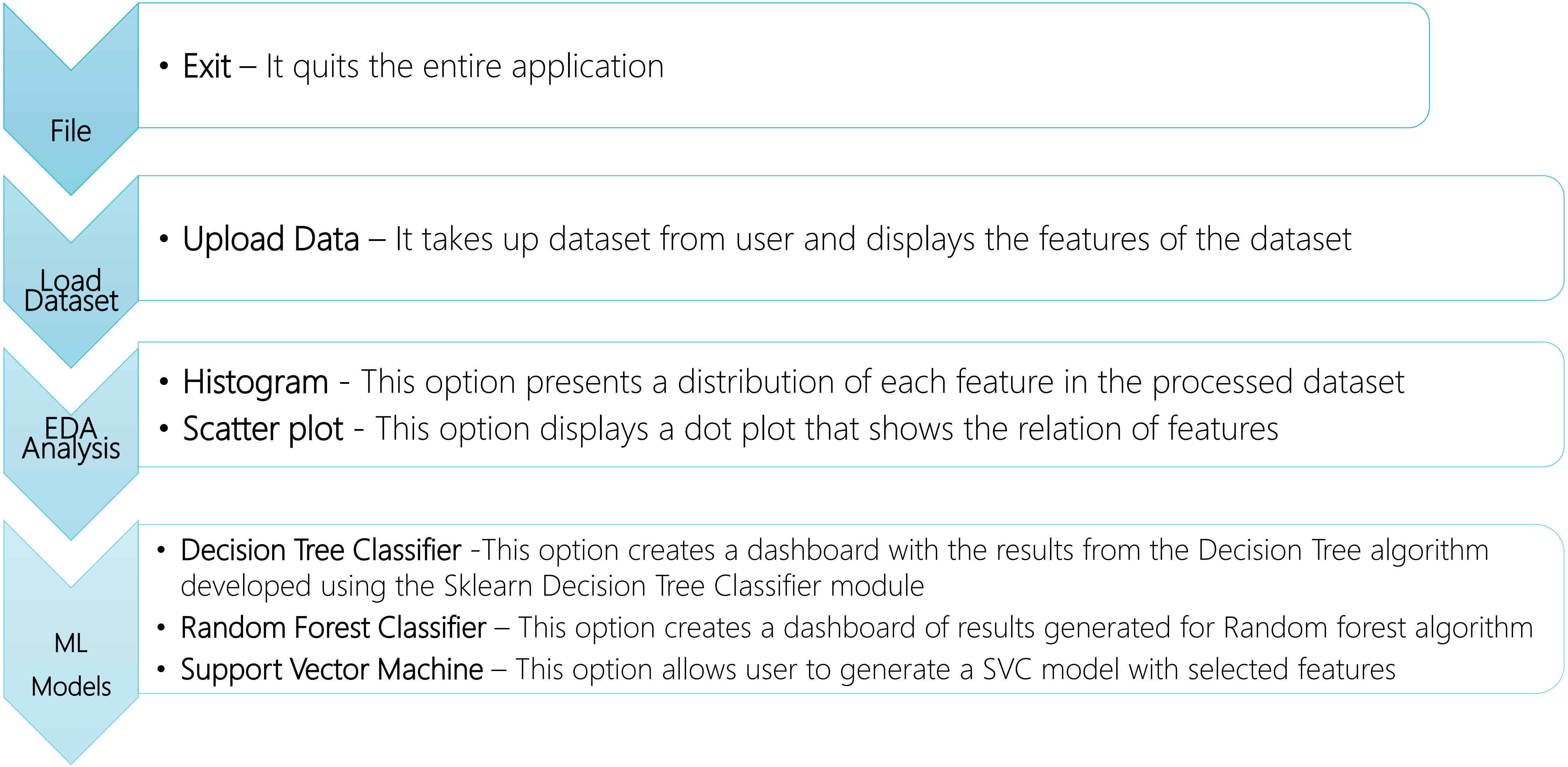
Accuracy is the fraction of predictions our model got right

Performance Measure

Decision tree is visualized by using Graphviz



Structure of the Application



File

- **Exit** – It quits the entire application

Load Dataset

- **Upload Data** – It takes up dataset from user and displays the features of the dataset

EDA Analysis

- **Histogram** - This option presents a distribution of each feature in the processed dataset
- **Scatter plot** - This option displays a dot plot that shows the relation of features

ML Models

- **Decision Tree Classifier** -This option creates a dashboard with the results from the Decision Tree algorithm developed using the Sklearn Decision Tree Classifier module
- **Random Forest Classifier** – This option creates a dashboard of results generated for Random forest algorithm
- **Support Vector Machine** – This option allows user to generate a SVC model with selected features

Results

Decision Tree Classifier:

- Accuracy of model = 77.6% (Test size=30%, Max_depth=3)
- ROC_AUC value = 73.01
- The gini and entropy models have similar accuracy

Random Forest Classifier:

- Accuracy of model = 75.80% (Test size=30, No. of estimators = 10, Criterion = Gini)
- ROC_AUC value = 73.03
- Gini model has better accuracy than the entropy model

Support Vector Classifier:

- Accuracy of model = 74.65%
- ROC_AUC value = 71.64

Conclusion

- Comparing the results of models, almost all the three models has accuracy value more than 70%
- Decision Tree Classifier tops the list by having the highest accuracy of 77%
- The decision tree and random forest models suffer when their parameter values like depth and estimators are changed
- The models in future enhancement needs to be tuned to predict the job change class correctly

Demo

Video link:
https://drive.google.com/file/d/1Y_u4un0_inFmVXfVGb4GQ2kmbNNgqJ7d/view?usp=sharing

Any Questions?

GitHub Link: https://github.com/adingankar/FINAL_PROJECT_GROUP7