Introduction to Business Analytics

Capstone Project Report Group 2

Group member:

Nguyen Huy Hoang 20194433

Hoang Thien Tam 20194450

Tran Quoc Khanh 20190085

Pham Nhu Thuan 20194456

1. Introduction

In recent technological decades, human recourse is becoming more and more valuable property of one company. Engineers and experts are essential for their operation and innovation. However, having a suitable strategy to manage human resources is challenging, especially for large incorporations. Employees are always seeking better working places.

In this project, we provide a solution by examining an Employee Attrition dataset. This dataset contains basic information about employees such as their age, distance from home, education, etc., and their attrition, quit the job or not. Data is analyzed by EDA techniques and visualized with popular plots, e.g. histogram, heatmap. Machine learning model is also applied to predict the quitting decision of employees, based on their characteristics.

Furthermore, in order to operate with real large-scale data, big data processing framework, Hadoop and Spark in our case, is used. Hadoop is well-organized for storing data and Spark is an efficient tool to process data, especially iterative operation. Thus, it is suitable for EDA and machine learning.

Based on that agenda, the second part is about setting up Hadoop and Spark in 3-node cluster, as well as instruction to manage data. Next, the third part is explanatory data analysis, machine learning in fourth section and final section is about conclusion and further development.

2. Hadoop and Spark

2.1 Set up hadoop hdfs.

We follow the guide to set up Hadoop 3 nodes cluster at:
https://phoenixnap.com/kb/install-hadoop-ubuntu

- All the installation steps and configuration setup are done the same on all three physical computer.
- First, we download and unpack hadoop:

wget https://downloads.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz

tar xzf hadoop-3.3.4.tar.gz

- Then we configure Environment Variables (bashrc) to add the following.

sudo nano .bashrc

```
export HADOOP_HOME=/home/hadoop/hadoop-3.3.4
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar

export SPARK_HOME=/home/hadoop/spark
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin

export PYSPARK_PYTHON=/usr/bin/python3
```

- Next, we edit /etc/hosts to add the IPs of all 3 nodes, configurationon node-master for example:

```
127.0.0.1       localhost
#127.0.1.1      tam-HP-ZBook-15-G3
192.168.39.139  tam-HP-ZBook-15-G3

192.168.39.139  node-master
192.168.39.137  hoang
192.168.39.56   khanh

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- Edit hadoop-env.sh File:

`sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh`

and add the line:

`export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre`

- Edit core-site.xml File

`sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml`

and add the following:

```
<configuration>
        <property>
            <name>fs.default.name</name>
            <value>hdfs://node-master:9000</value>
        </property>
        <property>
            <name>fs.defaultFS</name>
            <value>hdfs://node-master:8020</value>
        </property>

</configuration>
```

- Edit worker file:

`sudo nano $HADOOP_HOME/etc/hadoop/workers`

and add the following corresponding to .bashrc above, example in our case is:

```
hoang
node-master
khanh
```

- Edit hdfs-site.xml File

`sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml`

and add the following:

```
<configuration>
    <property>
            <name>dfs.namenode.name.dir</name>
            <value>/home/hadoop/data/nameNode</value>
    </property>

    <property>
            <name>dfs.datanode.data.dir</name>
            <value>/home/hadoop/data/dataNode</value>
    </property>

    <property>
            <name>dfs.replication</name>
            <value>2</value>
    </property>
</configuration>
```

- Edit mapred-site.xml File

`sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml`

and add the following:

```
<configuration>
    <property>
            <name>mapreduce.framework.name</name>
            <value>yarn</value>
    </property>
    <property>
            <name>yarn.app.mapreduce.am.env</name>
            <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
            <name>mapreduce.map.env</name>
            <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
            <name>mapreduce.reduce.env</name>
            <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
<property>
        <name>yarn.app.mapreduce.am.resource.mb</name>
        <value>512</value>
</property>

<property>
        <name>mapreduce.map.memory.mb</name>
        <value>256</value>
</property>

<property>
        <name>mapreduce.reduce.memory.mb</name>
        <value>256</value>
</property>
</configuration>
```

- Edit yarn-site.xml File

`sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml`

and add the following:

```
<configuration>

<!-- Site specific YARN configuration properties -->
    <property>
            <name>yarn.acl.enable</name>
            <value>0</value>
    </property>

    <property>
            <name>yarn.resourcemanager.hostname</name>
            <value>node-master</value>
    </property>

    <property>
            <name>yarn.nodemanager.aux-services</name>
            <value>mapreduce_shuffle</value>
    </property>

<property>
        <name>yarn.nodemanager.resource.memory-mb</name>
        <value>1536</value>
</property>

<property>
        <name>yarn.scheduler.maximum-allocation-mb</name>
        <value>1536</value>
</property>

<property>
        <name>yarn.scheduler.minimum-allocation-mb</name>
        <value>128</value>
</property>

<property>
        <name>yarn.nodemanager.vmem-check-enabled</name>
        <value>false</value>
</property>
<property>
  <name>yarn.scheduler.capacity.maximum-am-resource-percent</name>
  <value>100</value>
</property>
</configuration>
```

- Format HDFS NameNode

`hdfs namenode -format`

- Finally, we start all hdfs service with `start-all.sh` and put data using `hdfs dfs -put /path/to/local/file /path/to/hdfs/file`
- Check the file in hdfs using hadoop api in port: http://node-master:9870

2.2 Set up a spark cluster.

We follow the step to install spark in this link: https://phoenixnap.com/kb/install-spark-on-ubuntu

- First, we download and unpack spark using:

`wget https://downloads.apache.org/spark/spark-3.3.1/spark-3.3.1-bin-hadoop3.tgz`

`tar xvf spark-*`

- Move the spark folder to corresponding place according to .bashrc.

`sudo mv spark-3.3.1-bin-hadoop3 /home/hadoop/spark`

- We configure spark as follow:

`$SPARK_HOME/conf/spark-defaults.conf`, add the line:

`spark.master yarn`

`$SPARK_HOME/conf/spark-env.sh`

```
SPARK_MASTER_HOST='node-master'
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

`$SPARK_HOME/conf/workers`

```
hoang
node-master
khanh
```

- Finally, we start all using `$SPARK_HOME/sbin/start-all.sh`.

3. Explanatory Data Analysis

3.1 Overview of the dataset

In this project, we use IBM HR Analytics Employee Attrition & Performance dataset, which is available on Kaggle. This dataset contains 1470 samples, thirty-five attributes and no missing value. Our dataset is also imbalanced, there are more No on attrition than Yes, 1233 and 237 samples correspondingly.

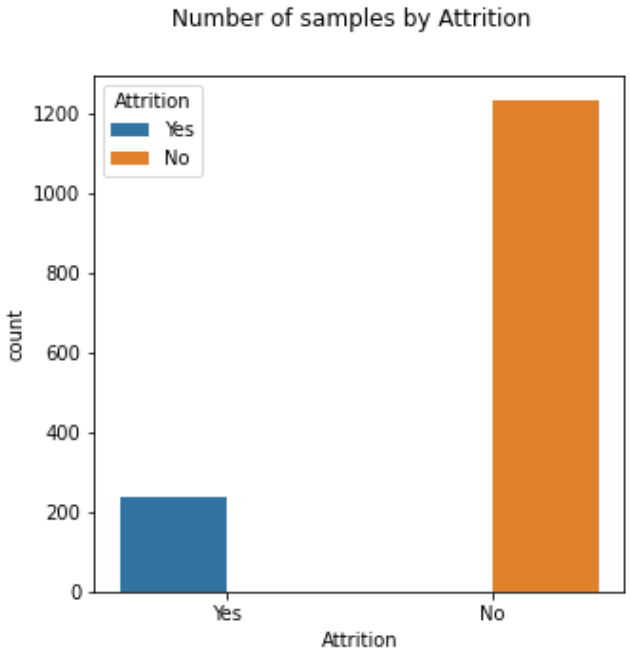

Figure 1 Number of samples by Attrtion

Among attributes, twenty-six of them have numeric format and nine have string format, although the types of attributes are not defined in that way, e.g. some ordinal variables have numeric format. Thus, for better analysis, if a numeric variable has few unique values and narrow range, we consider it as discrete continuous. Some ordinal columns also have numeric format, and their value is ambiguous, their definition is available on data webpage.

In the dataset, EmployeeNumber is the identified number, it is differed by employee. And all of employees are over 18, have the same EmployeeCount and StandardHours. Therefore, we ignore these attributes, EmployeeNumber, Over18, StandardHours and EmployeeCount.

3.2 Correlation

To evaluate the relationship between variables in the dataset, we use spearman correlation, since it can be used to both numeric and ordinal variables. It is also better in monotonicity detection of two variables.

$$\rho(X,Y) = 1 - 6\frac{\sum_{i=1}^{N}\left(r(X_i) - r(Y_i)\right)^2}{N^3 - N}$$

The Spearman formulation is formulated above. N is number of samples on dataset, $X\ and\ Y$ are pair of variables, $X_i and\ Y_i$ represent value of attributes at i^{th} row. $r(X_i)$ denotes rank of X_i value among variable X.

Correlation is visualized in the form of heatmap. For better observation, the upper part of the heatmap is masked.

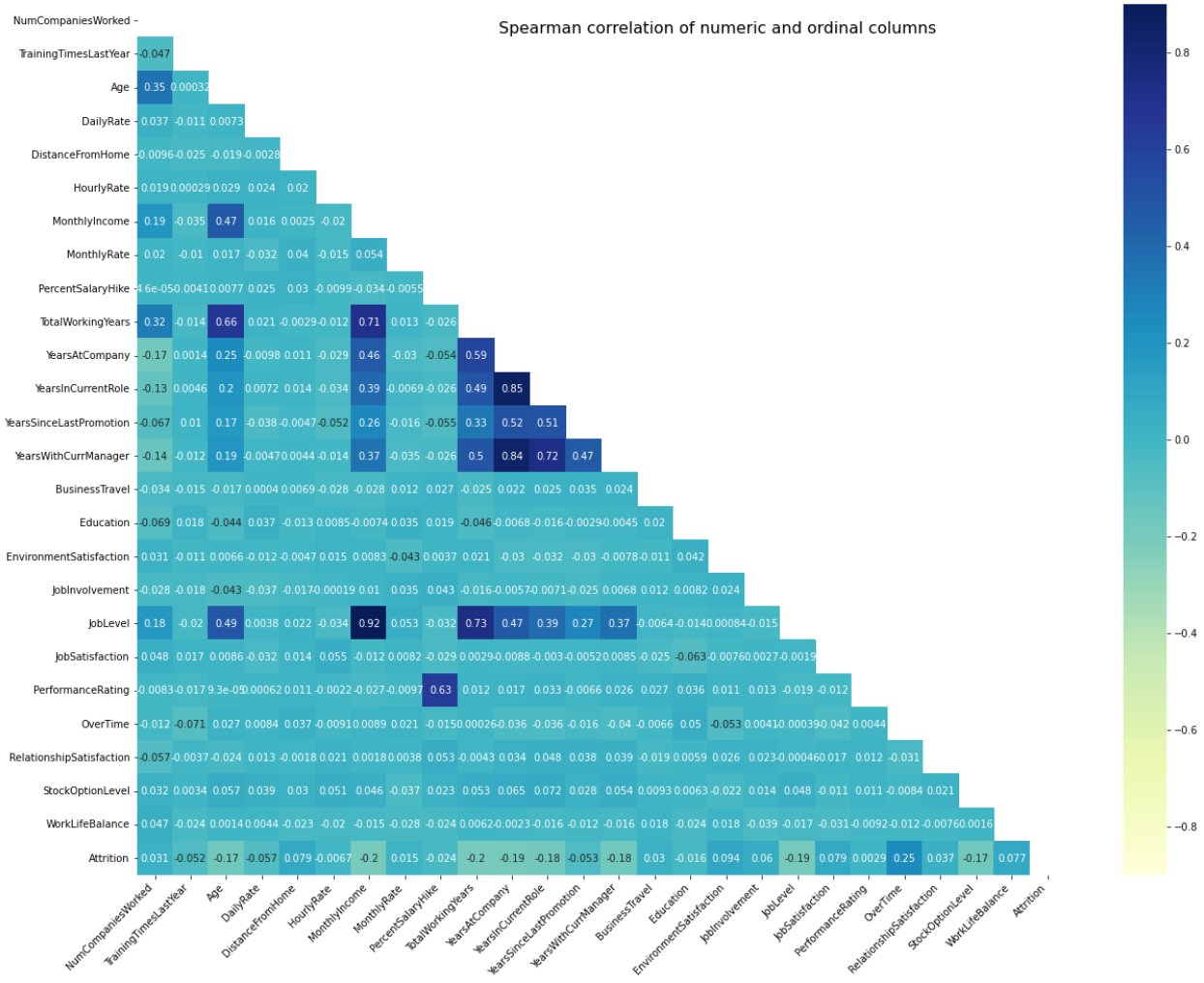

Figure 2 Spearman correlation heatmap of numeric and ordinal variables

The first observation is a group of variables which have strong correlation, TotalWorkingYears, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager, JobLevel and MonthlyIncome. This phenomenon is explainable. The more years employees have worked, the more years at the company, in current role and with current manager they have. It is also harder to be promoted when the total working year is high. Perhaps they have reached the highest position at the company.

MonthlyIncome is highly correlated with JobLevel. The higher-level employees are, the more they are paid. It also has a high correlation with TotalWorkingYears. An employee which has more experience is in high level of job and has large income.

About the target variables, there are no clear correlations between Attrition and other variables. However, it is slightly proportionally correlated with OverTime. The more time employees must work, the more likely they will quit. They seem to be tired and exhausted. On the other direction, Age, MonthlyIncome, TotalWorkingYears, YearsAtCompany YearsInCurrRole, and JobLevel are oppositely proportionally correlated. Possibly, old employees are more afraid of change, they have large income and high level, which make them less likely to quit their current job.

3.3 Data visualization

3.3.1 Numeric attributes

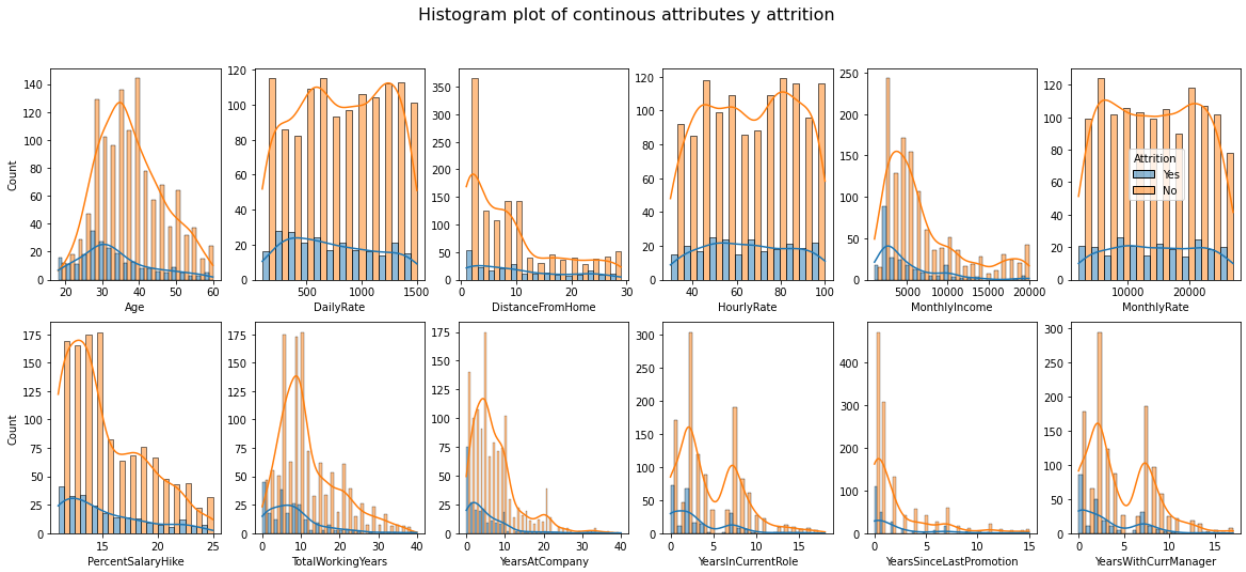

Figure 3 Histogram with kde plot of numeric variables, grouped by attrition.

For numeric attributes, histogram is kind of plot which discretizes variable value into range and count the number of samples fall on each range. The approximated distribution is also applied. We grouped data by Attrition to see whether there is difference in distribution of employee dropout by Attrition. Age, MothlyIncome and TotalWorkingYears seem to follow Gaussian, while DailyRate, HourlyRate and MonthlyRate are quite similar to uniform distribution.

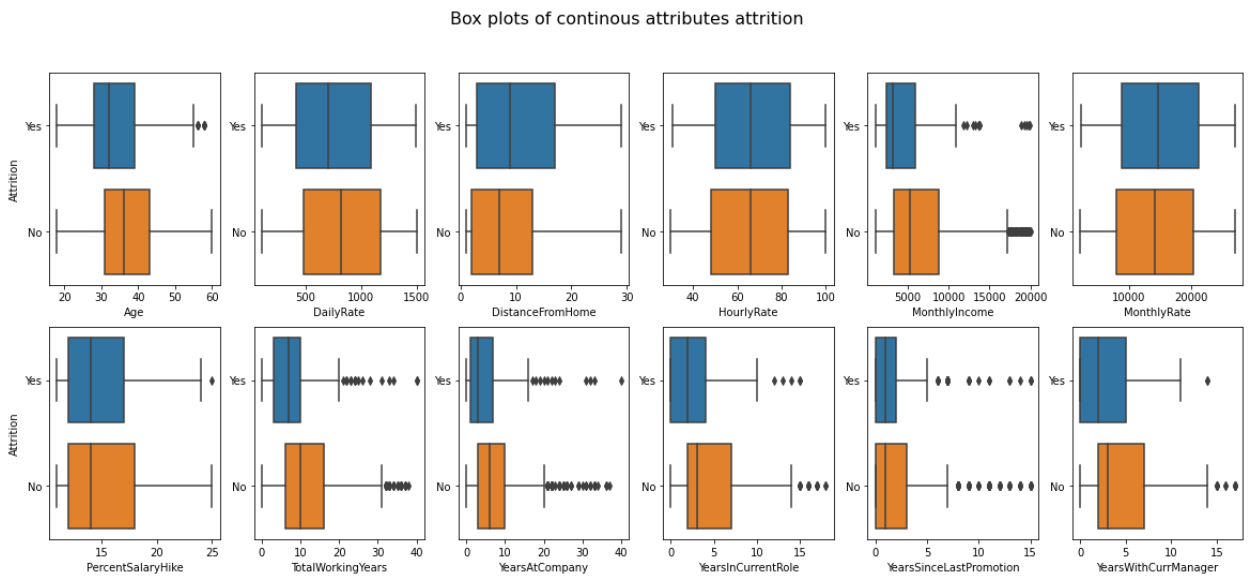

Figure 4 Box plot of numeric variables

Box plot is an appropriate visualization to see its distribution as well as exited outliers. Figure is the box plot of numeric attributes on the data, they are grouped by Attrition. MonthlyIncome, TotalWorkingYears, YearsAtCompany, YearsInCurrentRole and YearSinceLastPromotion are the attributes which have many outliers. We also can see noticeable differences in mean and interquartile range by Attrition, of DailyRate, DistanceFromHom, MonthlyIncome, TotalWorkingYears, YearAtCompany, YearsInCuttentRole and YearWithCurrManager.

3.3.2 Categorical variables

Histogram is also suitable for ordinal variables, below is the bar plot which counts number of occurrences, grouped by Attrition. Attrition rate is also plotted by the blue line. Left y-axis represents the count and right one is attrition rate. There are notable declining trends in attrition rate of attributes JobInvolvement and JobLevel. Joc drop-out percentage of Education, JobSatisfaction, RelationshipSatisfaction and WorkLifeBalance also reduces slightly when attributes value increases.

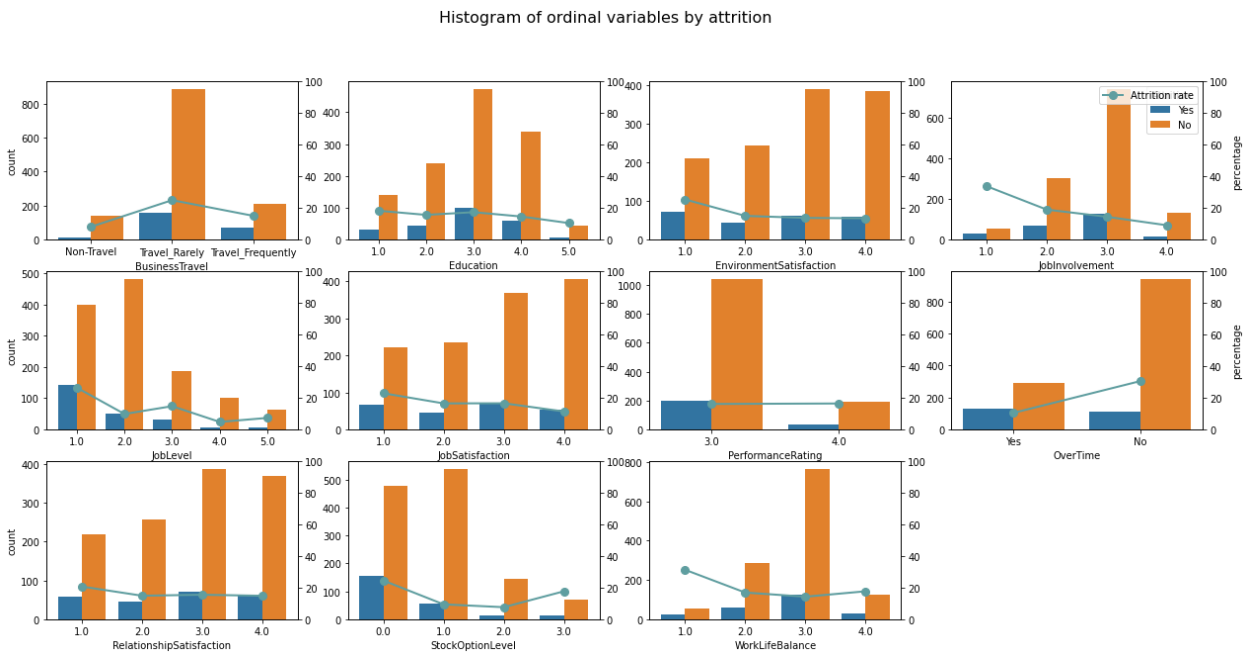

Figure 5 Histogram of ordinal variables grouped by Attrition, the line is percentage of Yes class per total

In order to examine the nominal variables, histogram and percentage stacked bar chart are used. Observing this chart, single employees have both the largest number of attritions and attrition rate, whereas divorced is the smallest. Regarding Gender attributes, the dropout rate of male and female are almost equal.

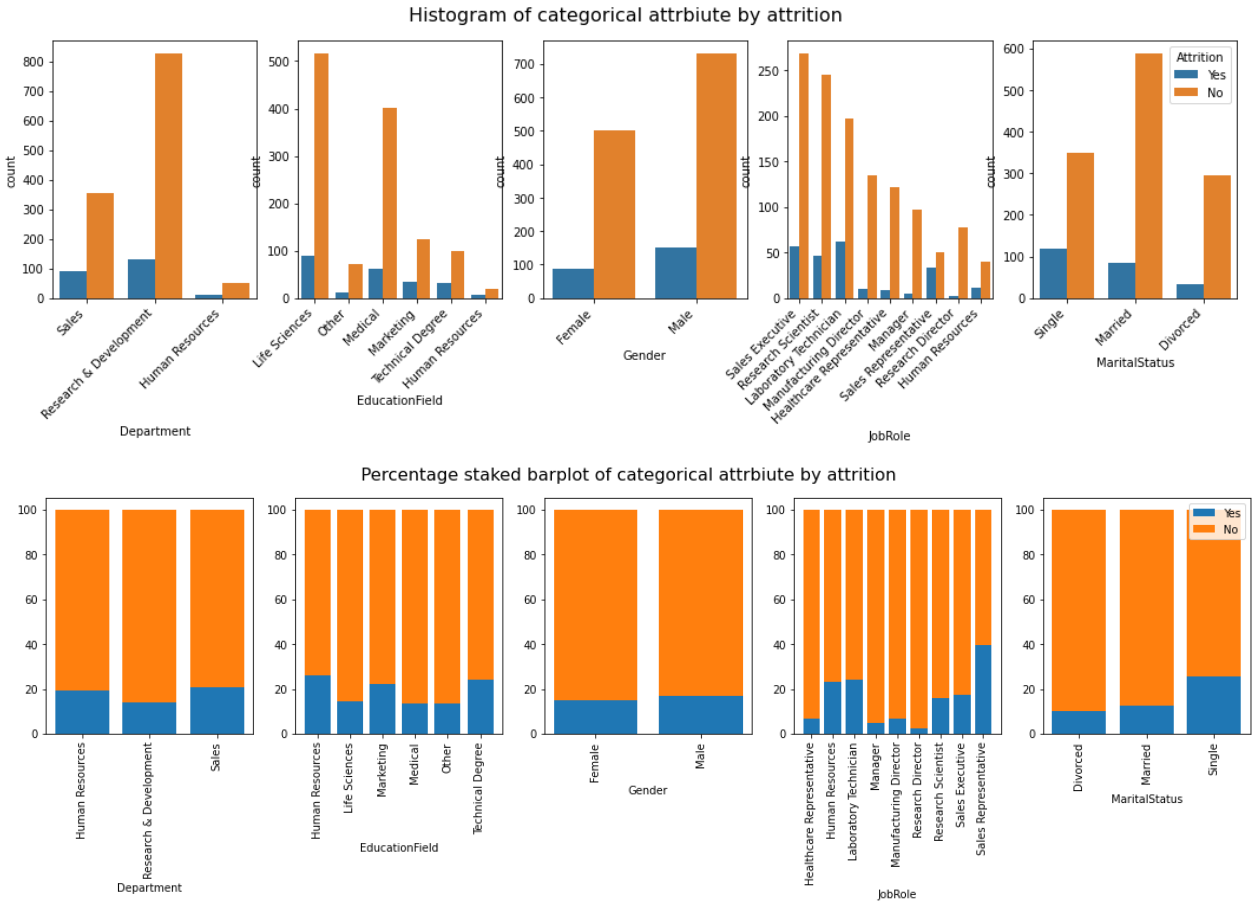

Figure 6 Above: histogram of nominal variables. Below: percentage stacked bar chart. Both are grouped by Attrition.

4. Machine Learning

In order to manage human recourse, a company needs to have an appropriate compensation policy. They also would like to predict whether a specific employee supposes to quit a job. It is important for the company to have a suitable strategy in the labor market.

Applying machine learning to is a good approach to this problem. Since the dataset is not complex, there are thirty attributes , EmployeeNumber, Over18, StandardHours and EmployeeCount in original data is redundant, and one target variable, traditional Machine Learning is suitable and sufficient. It does not require high computational capacity for training and inference. In this project, we apply two algorithms, Decision Tree and Random Forest. They are both implemented by

Spark Machine Learning on clusters, three nodes in total. Therefore, we are able to scale up with large real-world data.

To evaluate model performance, random split strategy is applied. Eighty percent of the data is used for training and twenty percent for testing. Both Decision Tree and Random Forest have many hyper-parameters to be defined in the beginning. Grid search with cross-validation evaluation is a way to find the best set of parameters on a defined range. For that reason, performance of model with specific set of hyper-parameters is evaluated in training set by 5-fold cross validation.

Since Decision Tree is likely to overfit training data, the set of searched params is defined to tackle this issue. They are related to complexity of the tree, includes maxDepth, minInstancesPerNode, and maxBins. maxDepth is the maximum allowed depth of the tree. Min instances per leaf node is the minimum number of samples for a node to be a leaf node. If a node splitting makes a leaf has less than that number, it cannot be split further. And maxBins is the maximum bin to split a numeric attribute value into range. Similar to Decision Tree, set of searched parameters for Random Forest are number of trees, min instances per node and max depth.

The accuracy on test set of Decision Tree and Random Forest are 80.87% and 85.37% respectively. This performance might vary with different running, due to splitting dataset. Following is their confusion matrix and report table, which is needed to analyze further.

Although random forest model has higher accuracy in general, its recall score on Yes class is lower. Detecting all intended dropout employees is important, because companies would like to predict who will quit the job. However, random forest is more precise than decision tree, i.e. if model predict yes, it is more likely the employee will quit.

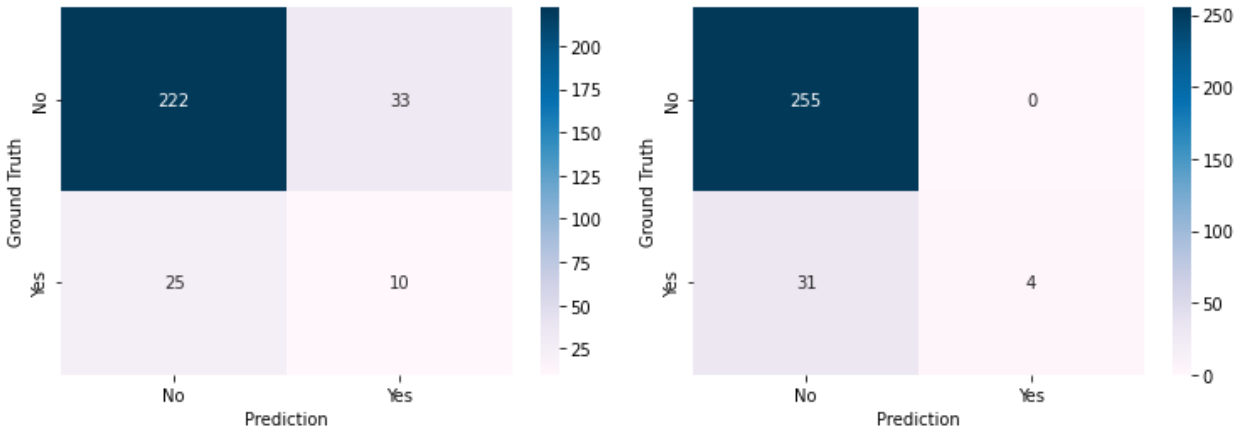

Figure 7 Confusion matrix of Decision Tree (left) and Random Forest (right)

Model	Attrition	Precision	Recall
Decision Tree	No	89.88%	87.06%
	Yes	23.26%	28.57%
Random Forest	No	89.16%	100%
	Yes	100%	11.42%

Table 1 Report table of Decision Tree and Random Forest

5. Conclusion and further development

To summary briefly, in this project, we have analyzed employe attrition based on their characteristics. In Explanatory Data Analysis part, we have analyzed attributes in the dataset. Regarding with target variables, Attrition, overtimed employees are more likely to quit, whereas old, high-level tend to stay longer. Furthermore, machine leaning models, Decision Tree and Random Forest, are applied to predict whether a specific employee will quit.

To make this project have the capacity to use in real life, big data processing tools like Hadoop and Spark, are also implemented. Big cooperations usually have tremendous number of employees, they will need efficient approaches to their problem.

Task	Responsible members
Hadoop installation, 3-node cluster setting up and data management	Nguyen Huy Hoang, Hoang Thien Tam, Pham Nhu Thuan
Spark installation and 3-node setting up	Hoang Thien Tam, Pham Nhu Thuan
Explanatory Data Analysis using Pyspark	Tran Quoc Khanh, Nguyen Huy Hoang

Machine Learning on Pyspark MLib	Tran Quoc Khanh, Pham Nhu Thuan
Report writing	Hoang Thien Tam, Tran Quoc Khanh

Table 2 Project tasks and responsible member

However, although the accuracy of machine learning model is quite high, around eighty percent, their performance in minor class is limited, due to the imbalance of dataset. To tackle this problem, we need to research more about imbalance dataset handling techniques, such as over and under sampling.

Moreover, explanatory data analysis is only focused on the relationship between pair of variables, while multi-variables analysis might be potential.

Source code is available at github.com/khanhtq2101/BusinessAnalysticsCapstoneProject