PREDICTING EMPLOYEE ATTRITION

Index

1. Introduction

EMPLOYEE TURNOVER has been identified as a key issue for organizations because of its adverse impact on work place productivity and long term growth strategies.

Attrition in human resources refers to the gradual loss of employees over time. In general, relatively high attrition is problematic for companies

The key to success in an organization is **the ability to attract and retain top talents**. It is vital for the Human Resource (HR) Department to identify the factors that keep employees and those which prompt them to leave.

PROBLEM:

COSTS

- Job posting
- Hiring processes paperwork
- New hire training

SOLUTION:

MACHINE LEARNING

2. Business context exploration and feature analysis

- 2.0 Loading Libraries and Dataset
- 2.1 Data Analysis and Cleaning
- 2.2 Features analysis

2. Business context exploration and feature analysis

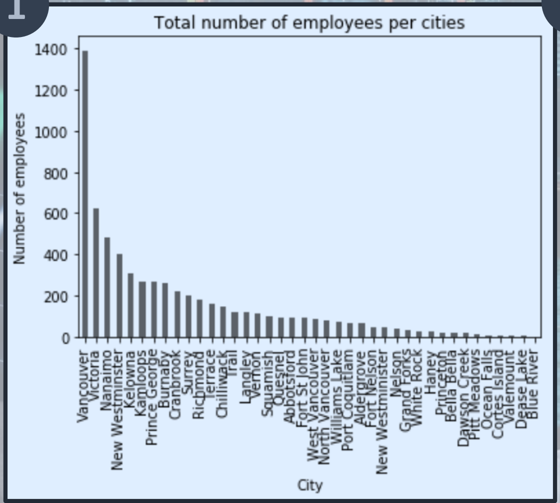
2.1 Data Analysis and Cleaning (1/3)

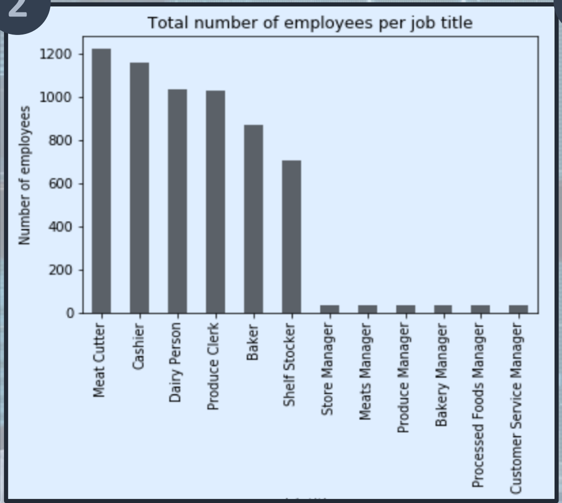
- Dimension of the dataset: (49653,18)
- Data types:

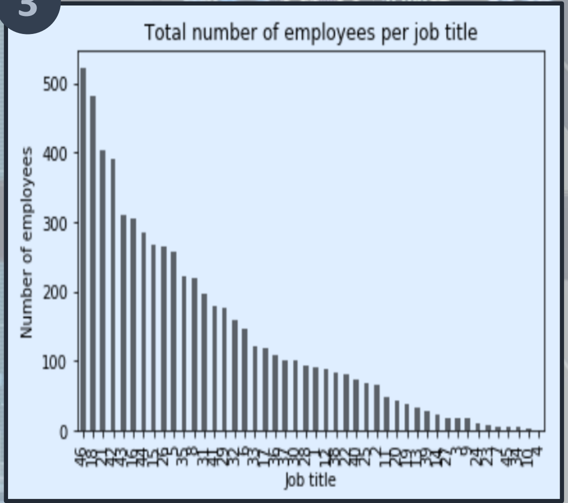
```
EmployeeID                     object
recorddate_key         datetime64[ns]
birthdate_key          datetime64[ns]
orighiredate_key       datetime64[ns]
terminationdate_key    datetime64[ns]
age                             int64
length_of_service               int64
city_name                      object
department_name                object
job_title                      object
store_name                     object
gender_short                   object
gender_full                    object
termreason_desc                object
termtype_desc                  object
STATUS_YEAR            datetime64[ns]
STATUS                         object
BUSINESS_UNIT                  object
dtype: object
```

- Employees:
  76% Active; 24% Inactive

- Some Preliminary barplots:
1. employees' distribution per department
2. employees' distribution per job title
3. employees' distribution per store name

1


2


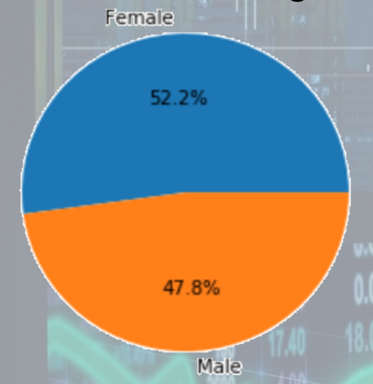
3


2. Business context exploration and feature analysis

2.1 Data Analysis and Cleaning (2/3)

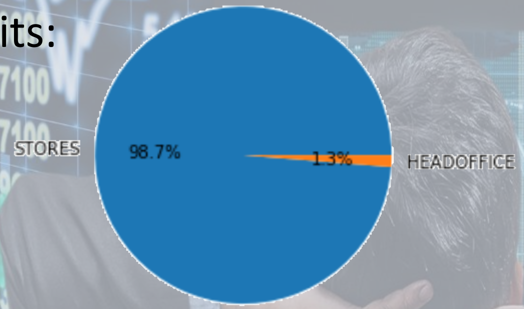
- Percentage of:
  48% Male; 52% Female

Gender Percentage:

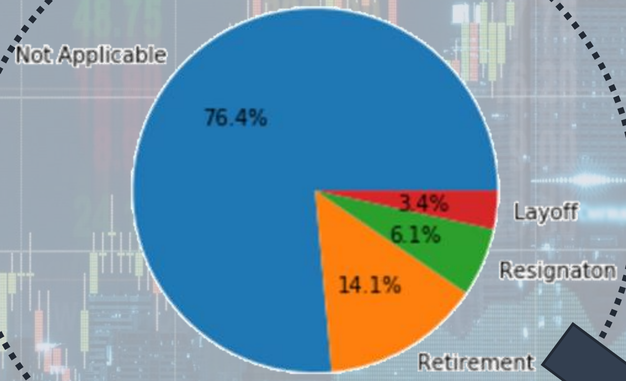


- Percentage of Business units:

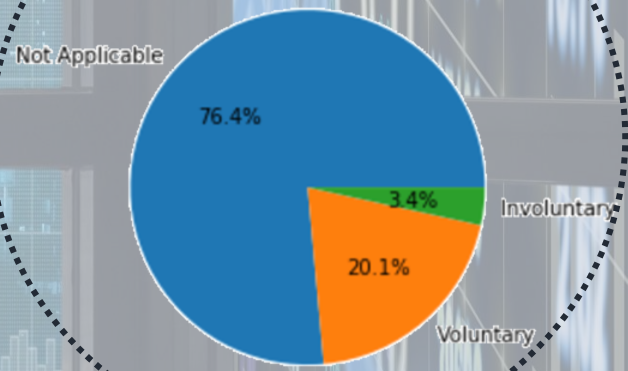
Business Units:



Reasons to leave:



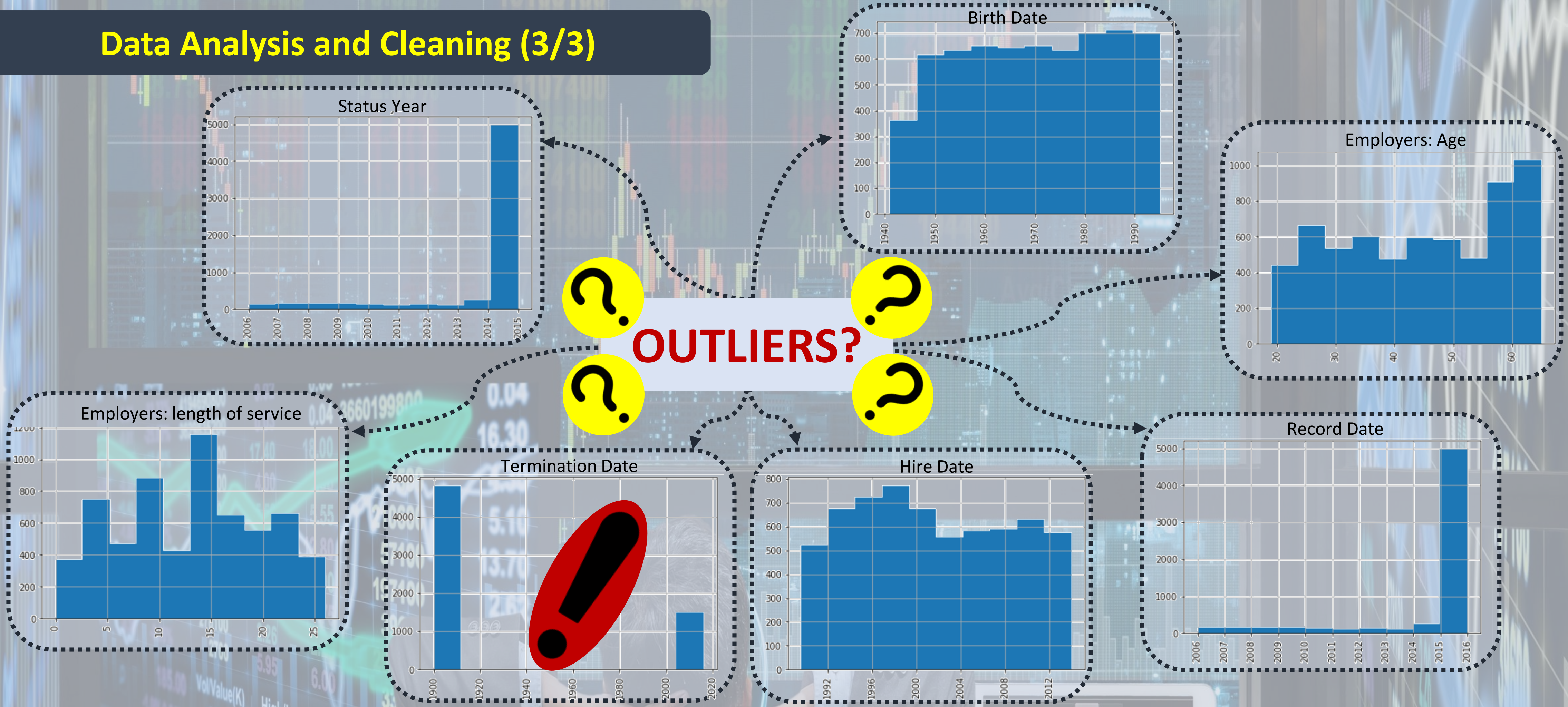
Type of reasons to leave:



2. Business context exploration and feature analysis

2.1

Data Analysis and Cleaning (3/3)



2. Business context exploration and feature analysis

2.2 Features analysis

Relevant Variables Resulted

	age	length_of_service	city_name	department_name	job_title	store_name	gender_full	STATUS
48424	65	13	Vernon	Meats	Meat Cutter	36	Female	0
48812	60	8	Vancouver	Meats	Meat Cutter	35	Male	0
48423	65	13	New Westminster	Meats	Meat Cutter	21	Female	0
48811	60	8	Richmond	Produce	Produce Clerk	29	Male	0
48808	60	8	Surrey	Produce	Produce Clerk	31	Male	0

3. Exploratory data analysis

- 3.1 Relationship between AGE and STATUS
- 3.2 Relationship between STATUS and LENGTH OF SERVICE
- 3.3 Relationship between STATUS and CITY NAME
- 3.4 Relationship between STATUS and DEPARTMENT NAME
- 3.5 Relationship between STATUS and JOB TITLE
- 3.6 Relationship between STATUS and STORE NAME
- 3.7 Relationship between STATUS and GENDER

3. Exploratory data analysis

Relevant Discoveries

3.1 The employees who leave the company are mainly **between 61-70 years old** (→ Retirement).

3.2 The employees who leave the company have generally **13 years of service** (→ Retirement).

3.3 This analysis is not that useful. Employees are mainly from the **city of Vancouver**, and so the active/unactive.

3.4 The employees who leave the most are from the **Meats department**.

3.5 Most of the active employees are cashiers, while most of the ex-employees are **Meat Cutters**.

3.6 The active employees are mainly from store 46. The employees who leave the most are from the **store 35**.

3.7 **Women** leave more than men.

3. Exploratory data analysis

4.1 **Relationship between STATUS and CITY NAME**

FOCUS

LOCATION of EMPLOYEES on an INTERACTIVE MAP

Despite the analysis is not that useful, for simple curiosity we wanted to graphically visualize where the employees are located on a map, just to give us a general idea and not only deal with tables and data.

1st tool used: ***"Foursquare" API***

From City, creation of Latitude and Longitude values

2nd tool used: ***"Folium" Package***

to plot 1st results

4. Model development

4.1 Train and Test Split

4.2 Imputation

4.3 Category Encoding

4.4 Feature Selection

4. Model development

4.1 Train and Test Split

1. 2.

```
4713 1571
0.7644812221514958
0.7644812221514958
```

3.

1. dimension of the training dataset
2. dimension of the testing dataset
3. propotions of the target variable in both of them

	age	length_of_service	city_name	department_name	job_title	store_name	gender_full	STATUS
count	4713.000000	4713.000000	4713	4713	4713	4713	4713	4713.000000
unique	NaN	NaN	39	21	41	45	2	NaN
top	NaN	NaN	Vancouver	Meats	Meat Cutter	46	Female	NaN
freq	NaN	NaN	1021	922	899	395	2472	NaN
mean	44.643751	12.803734	NaN	NaN	NaN	NaN	NaN	0.764481
std	14.116833	6.741237	NaN	NaN	NaN	NaN	NaN	0.424368
min	19.000000	0.000000	NaN	NaN	NaN	NaN	NaN	0.000000
25%	32.000000	7.000000	NaN	NaN	NaN	NaN	NaN	1.000000
50%	45.000000	13.000000	NaN	NaN	NaN	NaN	NaN	1.000000
75%	58.000000	19.000000	NaN	NaN	NaN	NaN	NaN	1.000000
max	65.000000	26.000000	NaN	NaN	NaN	NaN	NaN	1.000000

4.2 Imputation

There are no missing values in train and test set, so we don't need to do the imputation

	perc_miss
age	0.0
length_of_service	0.0
city_name	0.0
department_name	0.0
job_title	0.0

4. Model development

4.3 Category Encoding

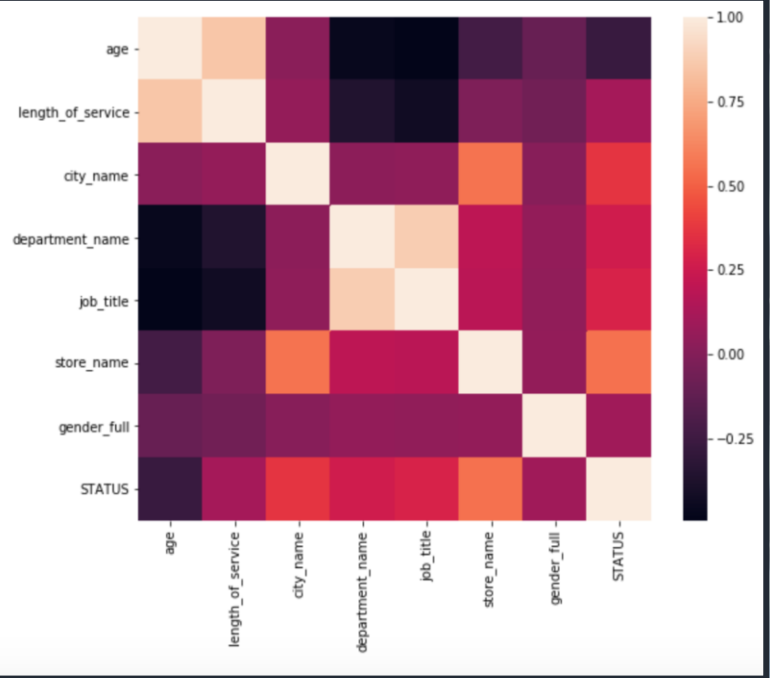
Target mean encoding method in order to have **numerical** columns .

```
train.shape
(4713, 8)
```

	age	length_of_service	city_name	department_name	job_title	store_name	gender_full	STATUS
count	4713.000000	4713.000000	4713.000000	4713.000000	4713.000000	4713.000000	4713.000000	4713.000000
mean	44.643751	12.803734	0.773344	0.770471	0.773033	0.775103	0.764473	0.764481
std	14.116833	6.741237	0.112084	0.090345	0.100855	0.199608	0.040533	0.424368
min	19.000000	0.000000	0.159267	0.266120	0.266120	0.100521	0.725884	0.000000
25%	32.000000	7.000000	0.766413	0.698117	0.706980	0.768597	0.725884	1.000000
50%	45.000000	13.000000	0.788210	0.778639	0.783874	0.818208	0.725884	1.000000
75%	58.000000	19.000000	0.818208	0.870312	0.893239	0.897114	0.807039	1.000000
max	65.000000	26.000000	0.864999	0.883469	0.898276	0.934831	0.807039	1.000000

4.4 Feature selection

Most of the variables are positively correlated to STATUS; *«City Name»* is **slightly positive** correlated to STATUS, instead of *«Age»* that is **negative** correlated to it.



5. Machine learning models

5.1 Logistic Regression

5.2 Decision Tree

5.3 Random Forest

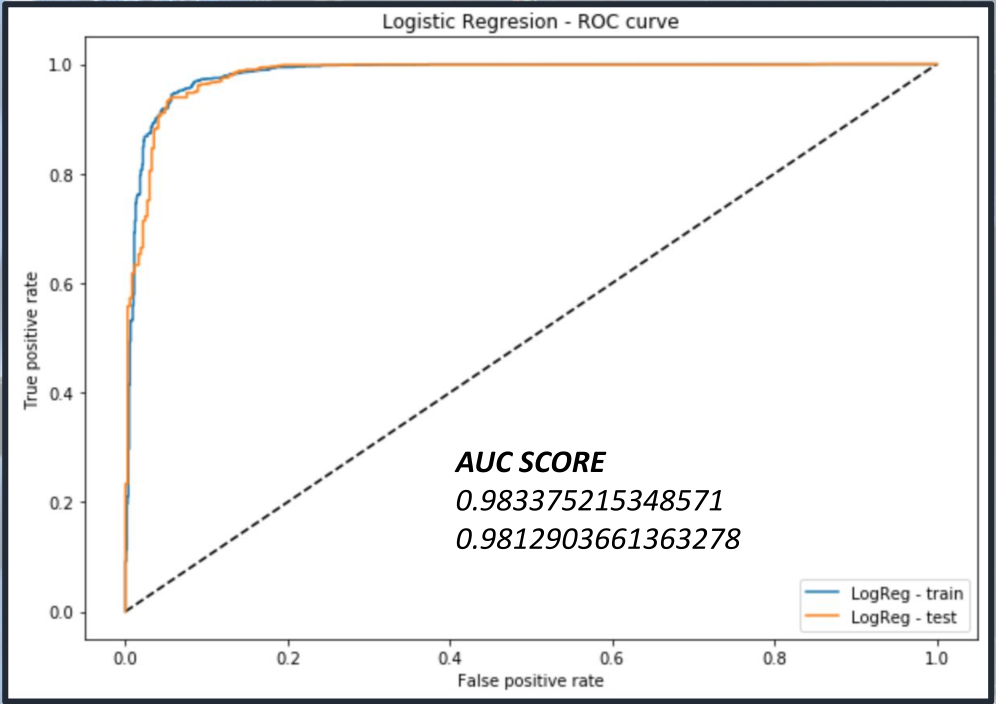
5.4 Gradient Boosting

5.5 Neural Network

5. Machine learning models

5.1 Logistic Regression

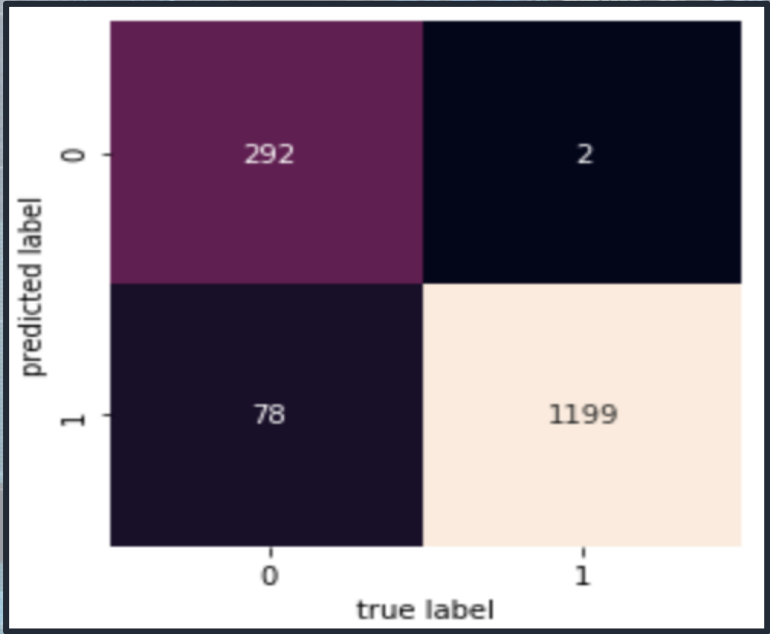
ROC CURVE FOR TRAIN AND TEST DATA



ROC is a probability curve and **AUC** represents a model's ability to discriminate between positive and negative classes.

Very good ROC curves. There is a slightly difference between train and test logistic regression. An area of 1.0 represents a model that made all predictions perfectly.

CONFUSION MATRIX



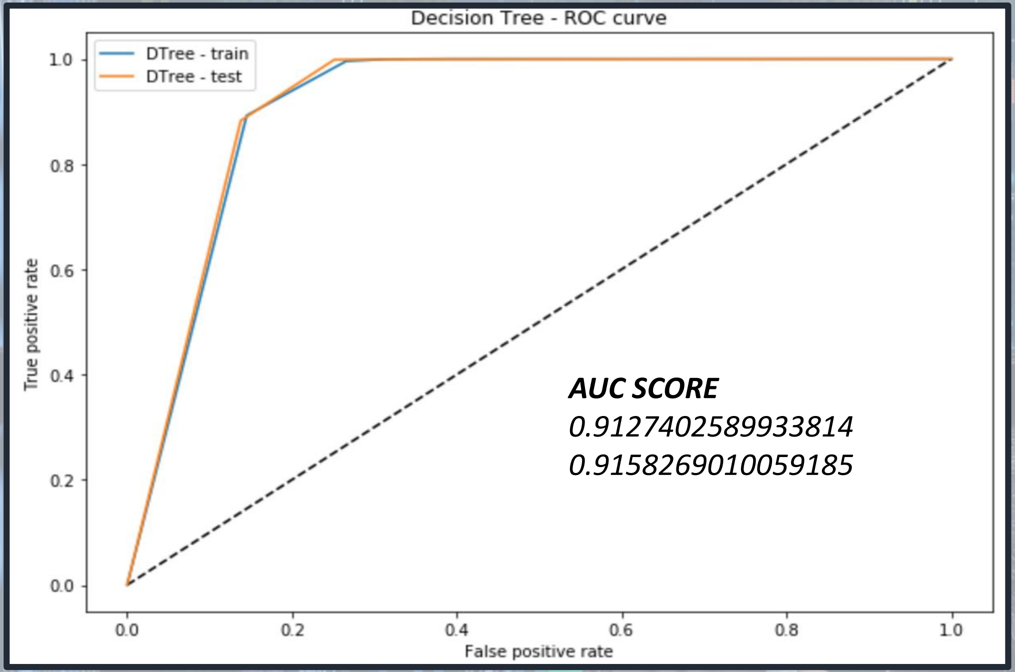
292+1199 = 1491 **correct** predictions
78+2 = 80 **incorrect** predictions

5. Machine learning models

5.2 Decision Tree

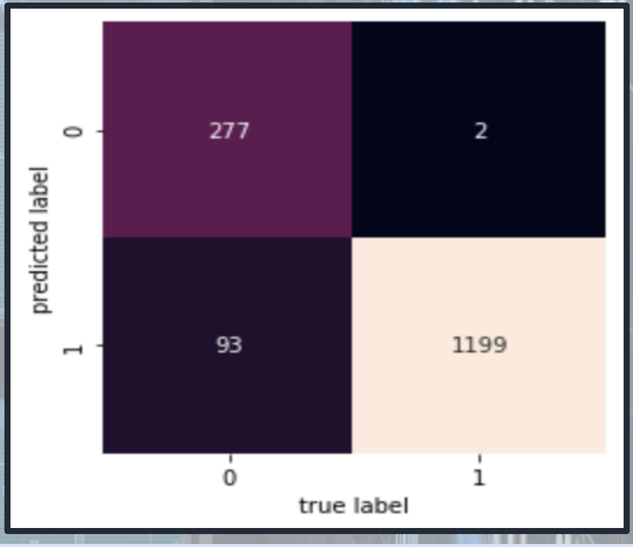
It is a predictive model which is a mapping from observations about an item to conclusions about its target value.

ROC CURVE FOR TRAIN AND TEST DATA



For the moment, we prefer **Logistic Regression** model since it has a higher AUC score compared to the **Decision Tree** model

CONFUSION MATRIX



277+1199 = 1476 **correct** predictions
93+2 = 95 **incorrect** predictions

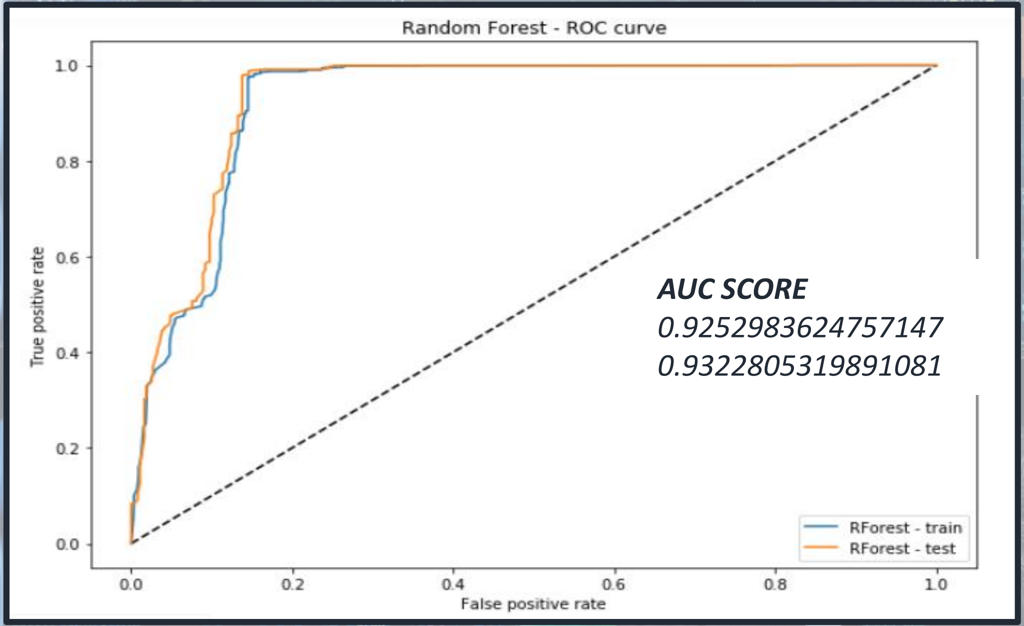
5. Machine learning models

5.3

Random Forest

Consists of a large number of individual decision trees that operate as an ensemble. Each individual tree spits out a class prediction and the class with the most votes becomes our model's prediction

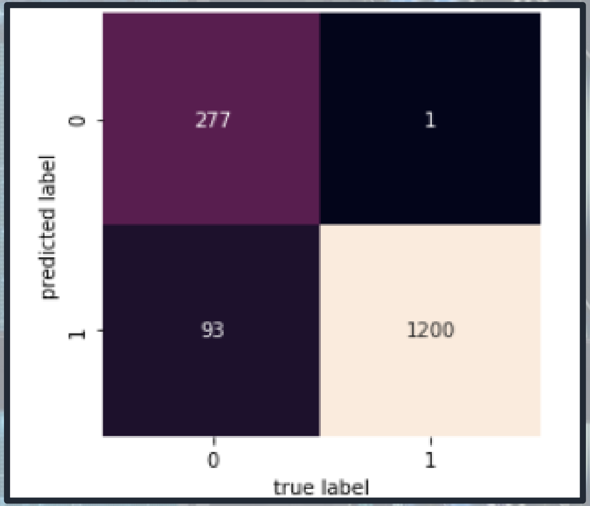
ROC CURVE FOR TRAIN AND TEST DATA



AUC SCORE
0.9252983624757147
0.9322805319891081

For the moment, we prefer **Logistic Regression** model since it has a higher AUC score compared to the **Random Forest** model

CONFUSION MATRIX



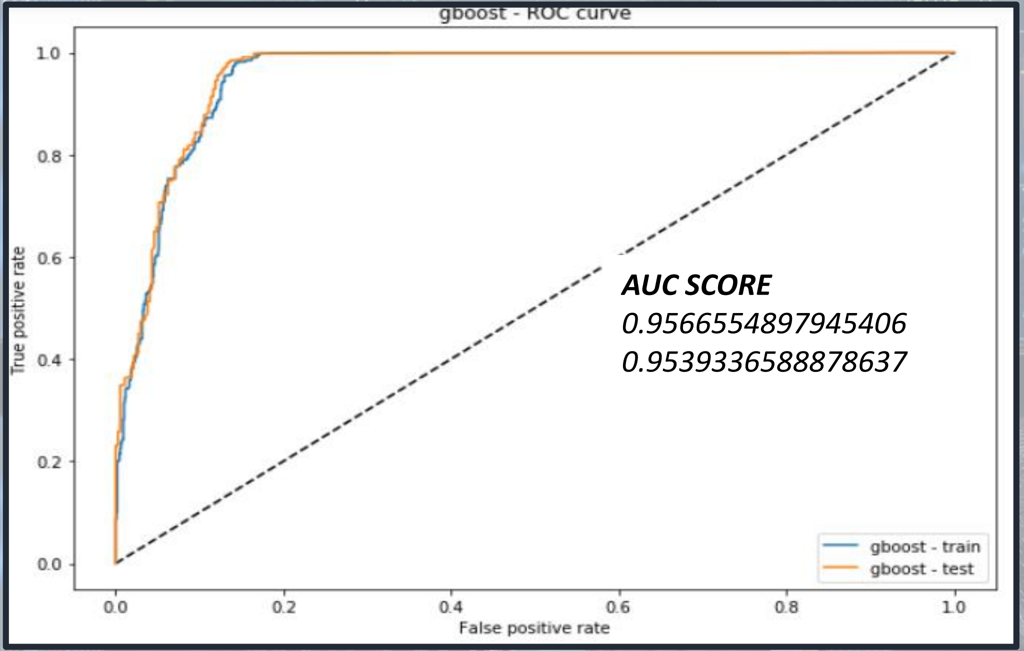
277+1200 = 1477 correct predictions
93+1 = 94 incorrect predictios

5. Machine learning models

5.4 Gradient Boosting

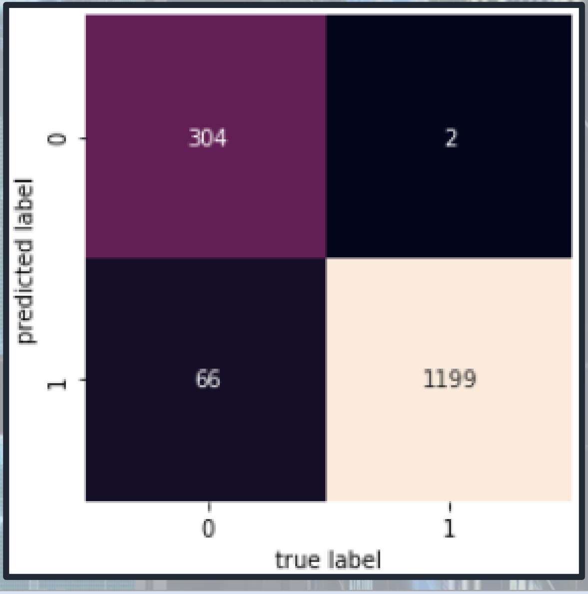
Technique for regression and classification problems. The best possible next model, when combined with previous models, minimizes the overall prediction error

ROC CURVE FOR TRAIN AND TEST DATA



For the moment, we prefer **Logistic Regression** model since it has a higher AUC score compared to the **Gradient Boosting** model

CONFUSION MATRIX



304+1199 = 1503 correct predictions
66+2 = 68 incorrect predictios

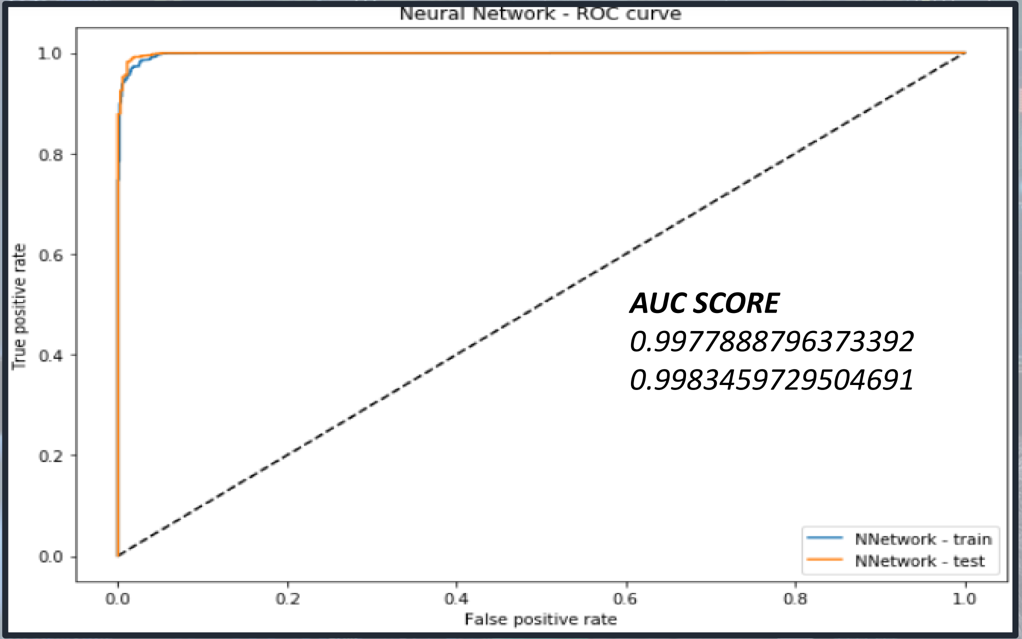
5. Machine learning models

5.5

Neural Network

Neural networks are a series of algorithms that mimic the operations of a human brain to recognize relationships between vast amounts of data.

ROC CURVE FOR TRAIN AND TEST DATA



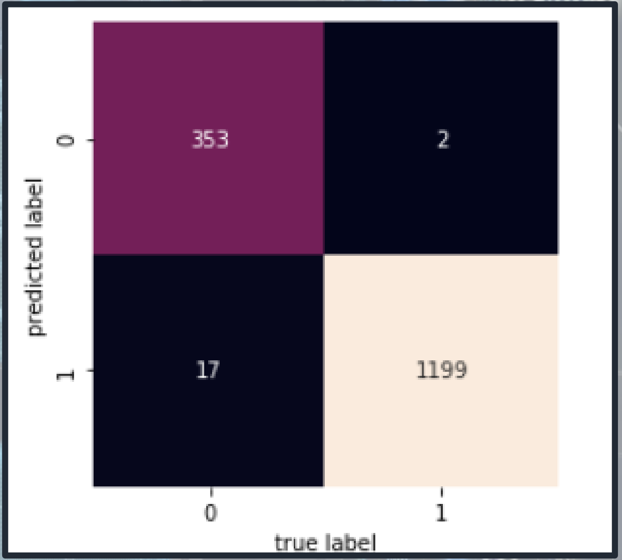
AUC SCORE

0.9977888796373392

0.9983459729504691

In conclusion, we prefer **Neural Network** model since it has a higher AUC score compared to the **Logistic Regression** model

CONFUSION MATRIX

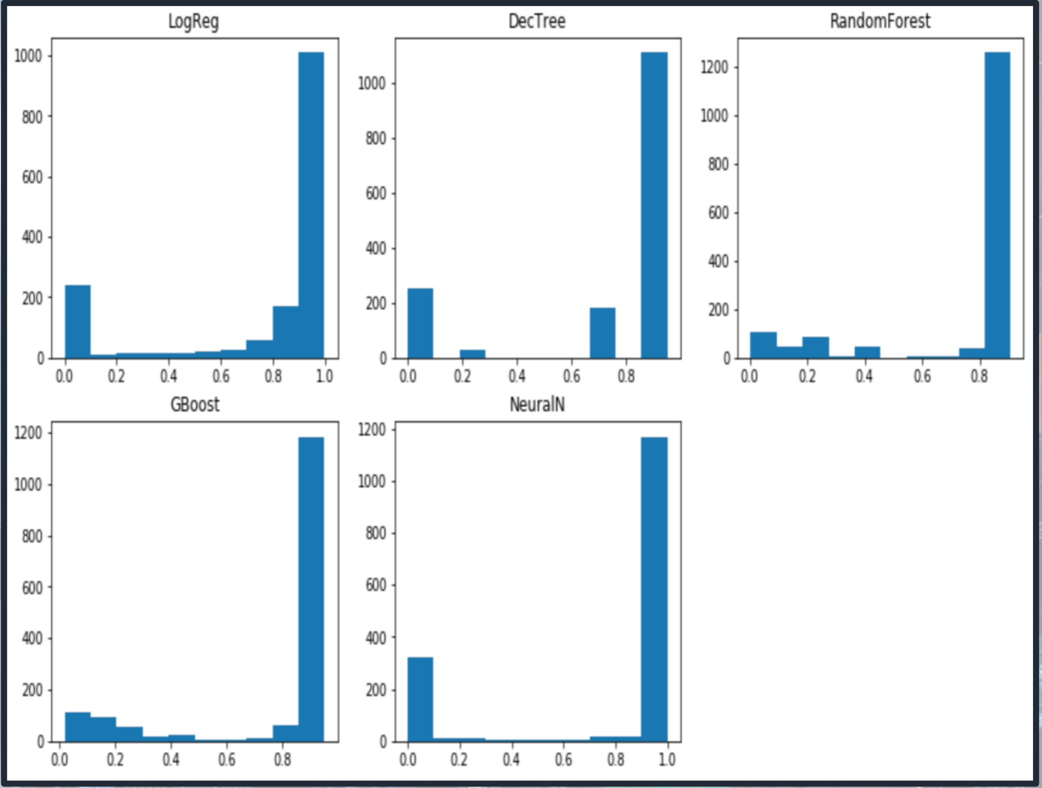


353+1199 = 1552 correct predictions

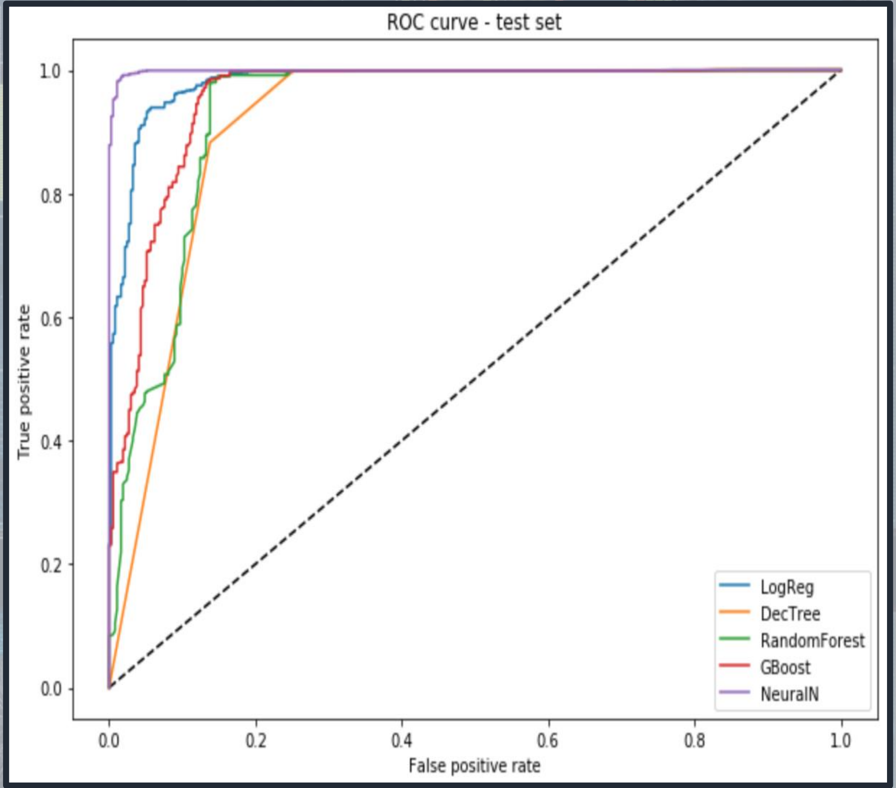
17+2 = 19 incorrect predictios

6. Model Comparison

Plot the probability distribution of all the models created



ROC CURVE SCORE



	train_ROC	test_ROC
Logistic Regression	0.983375	0.981290
Decision Tree	0.912740	0.915827
Random Forest	0.925298	0.932281
Gradient Boosting	0.953934	0.956655
Neural Network	0.997789	0.998346

As we said, we prefer **NEURAL NETWORK** since it has a higher AUC score compared to the other models.
The ROC curve is almost at 90 degrees indicating that the model performs very well

7. Model Results 1/3

Accuracy

Percentage of correctly classified records on the total

```
accuracy for lr_preds: 0.9490770210057289
accuracy for dt_preds: 0.939528962444303
accuracy for rf_preds: 0.9401654996817314
accuracy for gb_preds: 0.956715467854869 5
accuracy for nn_preds: 0.9879057924888606
```

Precision-Recall

Precision: TP/(TP+FP)
Recall: TP/(TP+FN)

```
precision for lr_preds: 0.9389193422083008
recall for lr_preds: 0.9983347210657785


precision for dt_preds: 0.9280185758513931
recall for dt_preds: 0.9983347210657785


precision for rf_preds: 0.9280742459396751
recall for rf_preds: 0.9991673605328892


precision for gb_preds: 0.9478260869565217
recall for gb_preds: 0.9983347210657785


precision for nn_preds: 0.9860197368421053
recall for nn_preds: 0.9983347210657785
```

F1 score

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

```
for lr_preds: 0.9677158999192899
for dt_preds: 0.9618933012434818
for rf_preds: 0.9623095429029671
for gb_preds: 0.9724249797242499
for nn_preds: 0.9921390153082333
```

7. Model Results 2/3

RESULTS with a different Cut-Off

We did our analysis using the default cut-off of 0.5.
To improve our model we would like to fix a different cut-off.

Because of the high accuracy, we can't just choose a value as cut-off but we have to choose **the best cut-off value**. The optimal cut off point is where the “true positive rate” is high and the “false positive rate” is low. The optimal cut-off is the point where there is the **elbow of the ROC curve: 0.7698501664999003**

LOGISTIC

	score
Accuracy	0.949714
Precision	0.963636
Recall	0.970858
F1-Score	0.967234

DECISION TREE

	score
Accuracy	0.877785
Precision	0.954095
Recall	0.882598
F1-Score	0.916955

RANDOM FOREST

	score
Accuracy	0.940165
Precision	0.929403
Recall	0.997502
F1-Score	0.962249

GRADIENT BOOSTING

	score
Accuracy	0.955442
Precision	0.954948
Recall	0.988343
F1-Score	0.971358

NEURAL NETWORK

	score
Accuracy	0.987906
Precision	0.994975
Recall	0.989176
F1-Score	0.992067

7. Model Results 3/3

Esembling models

Process of creating a model composed by different algorithms in order to gain a better prediction of the outcome. The goal of is to reduce the generalization error of the prediction.

We put together all the models except Decision tree (because it is has the smaller accuracy).

Scores for the esembling models:

```
train auc: 0.9970112493842719
test auc: 0.9964286517991763
train f1: 0.9741297575551131
test f1: 0.9771986970684039
train recall: 0.9980571745767416
test recall: 0.9991673605328892
train precision: 0.9513227513227513
test precision: 0.9561752988047809
```

Create a dataframe with the results:

	auc_test	auc_train	f1_test	f1_train	precision_test	precision_train	recall_test	recall_train
lr	0.981290	0.983375	0.967716	0.965536	0.938919	0.937516	0.998335	0.995282
dt	0.915827	0.912740	0.961893	0.958450	0.928019	0.924008	0.998335	0.995559
rf	0.932281	0.925298	0.962310	0.959840	0.928074	0.924203	0.999167	0.998335
gb	0.956655	0.953934	0.972425	0.971629	0.947826	0.946565	0.998335	0.998057
nn	0.998346	0.997789	0.992139	0.990765	0.986020	0.984118	0.998335	0.997502
ensemble	0.996429	0.997011	0.977199	0.974130	0.956175	0.951323	0.999167	0.998057

8. Conclusion

Goal: to find the best model to predict whether an employee would leave the company.

Target variable: status of an employee, which can be either active or not active.

With the ensemble model we did not gain an improvement in the prediction with respect to the Neural Network and we feel free to say that, in our case, the neural network is the **best choice** for the classification problem we were dealing with.

This model could bring to the firms money saving and time saving.