Individual Final Report
DATS 6103 Final project

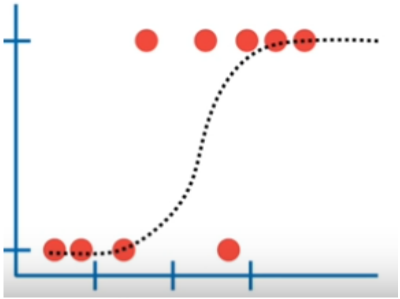
Yuchen Ma

Introduction

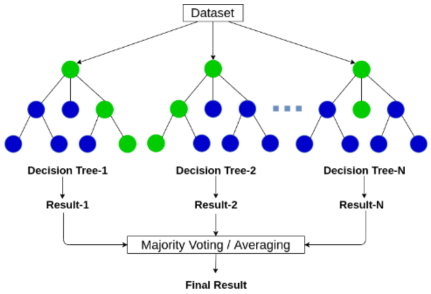
Data science is a new industry. Since it gets more popular in society, more companies need data scientists. The company which is active in Big Data and Data Science wants to hire data scientists among people who successfully pass some courses which are conducted by the company. Many people sign up for their training, but not all people will work for the company after training. Therefore, the company wants to know which of these candidates really wants to work for the company after training or looking for a new employment. That is the reason why we explore this topic. In this project, we aim to explore the probability of a candidate to look for a new job or will work for the company after training, as well as interpreting affected factors on employee decision. This project can help the company to reduce the cost and time as well as the quality of training.

Description of your individual work

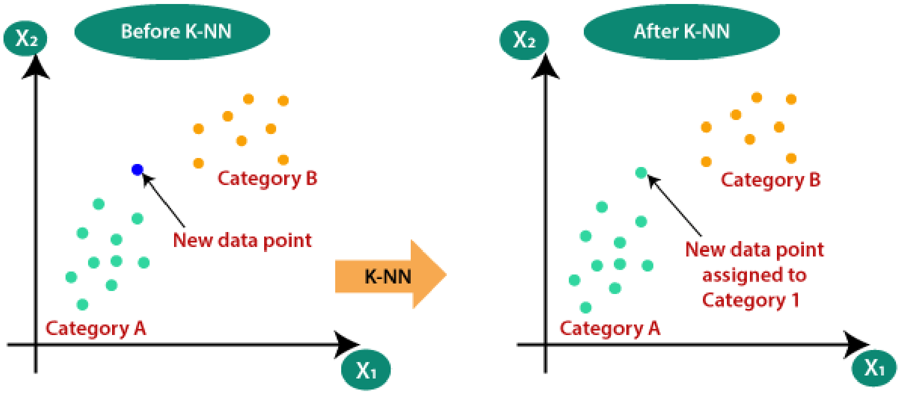
In this project, I use **logistic regression** which is a classification algorithm and is used to predict a binary outcome based on a set of independent variables.



I also use **random forest** which is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.



I also use K-Nearest Neighbor(KNN) algorithm. K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. The K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.



Describe the portion of the work that you did on the project in detail

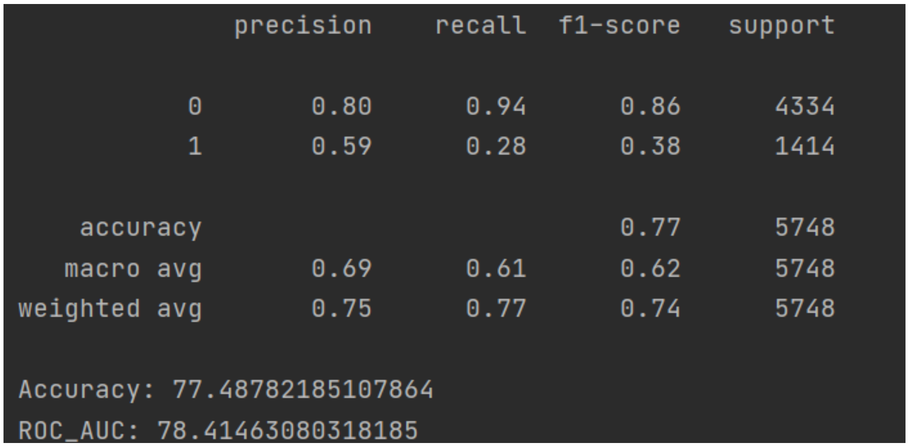
1. I build logistic regression model, random forest model and KNN model. Then I get ROC_AUC result and compare them.

2. In the final, I combine everyone's working together and make the final report.

```
## model
# label target variable
le = LabelEncoder()
y = le.fit_transform(y)

# split the dataset into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=2000)

# logistic regression model
lr = LogisticRegression()
lr.fit(X_train, y_train)
# Make predictions
y_pred1 = lr.predict(X_test)
y_pred_score1 = lr.predict_proba(X_test)
print(classification_report(y_test,y_pred1))
print("Accuracy:", accuracy_score(y_test, y_pred1) * 100)
print("ROC_AUC:", roc_auc_score(y_test,y_pred_score1[:,-1]) * 100)

# randomforest
clf = RandomForestClassifier(n_estimators=90)
clf.fit(X_train, y_train)
## Make predictions
y_pred = clf.predict(X_test)
y_pred_score = clf.predict_proba(X_test)
print(classification_report(y_test,y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred) * 100)
print("ROC_AUC:", roc_auc_score(y_test,y_pred_score[:,-1]) * 100)

# KNN
clf_KNN = KNeighborsClassifier(n_neighbors=5)
clf_KNN.fit(X_train, y_train)
#make prediction
y_pred0 = clf_KNN.predict(X_test)
y_pred_score0 = clf_KNN.predict_proba(X_test)

print(classification_report(y_test,y_pred0))
print("Accuracy:", accuracy_score(y_test, y_pred0) * 100)
print("ROC_AUC:", roc_auc_score(y_test,y_pred_score0[:,-1]) * 100)
```

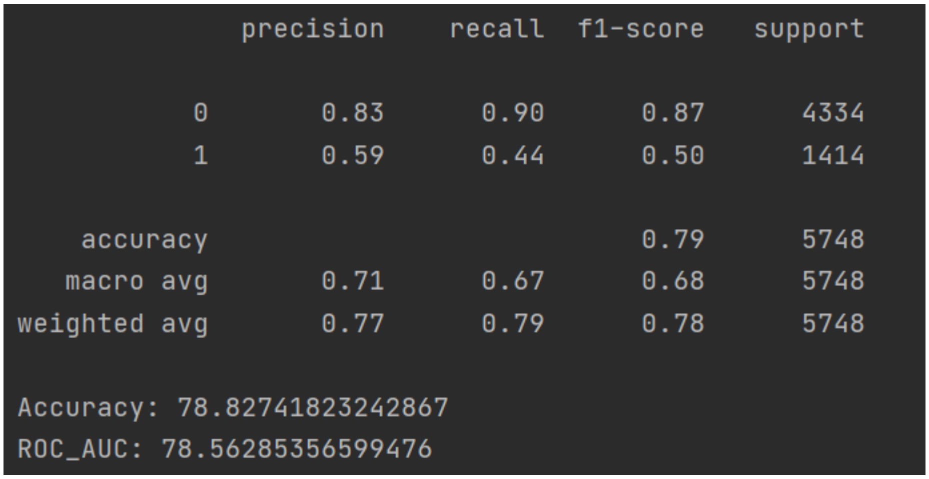
Result

This is result of logistic regression:



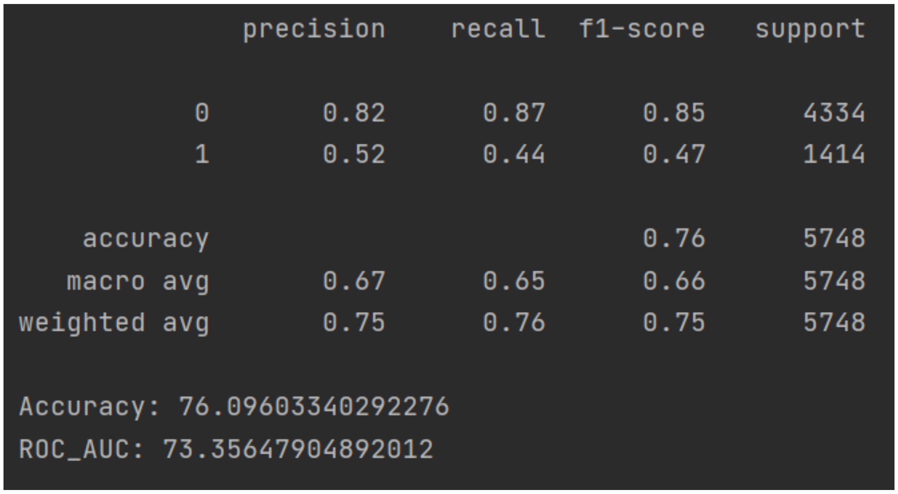
```
              precision    recall  f1-score   support

           0       0.80      0.94      0.86      4334
           1       0.59      0.28      0.38      1414

    accuracy                           0.77      5748
   macro avg       0.69      0.61      0.62      5748
weighted avg       0.75      0.77      0.74      5748

Accuracy: 77.48782185107864
ROC_AUC: 78.41463080318185
```

This is the result of random forest



```
              precision    recall  f1-score   support

           0       0.83      0.90      0.87      4334
           1       0.59      0.44      0.50      1414

    accuracy                           0.79      5748
   macro avg       0.71      0.67      0.68      5748
weighted avg       0.77      0.79      0.78      5748

Accuracy: 78.82741823242867
ROC_AUC: 78.56285356599476
```

This is the result of KNN



```
              precision    recall  f1-score   support

           0       0.82      0.87      0.85      4334
           1       0.52      0.44      0.47      1414

    accuracy                           0.76      5748
   macro avg       0.67      0.65      0.66      5748
weighted avg       0.75      0.76      0.75      5748

Accuracy: 76.09603340292276
ROC_AUC: 73.35647904892012
```

Summary and conclusions

Compare the value of accuracy of three models, we can find that using random forest can get highest accuracy, so we select random forest model and discard logistic regression model KNN model.

References

https://www.kaggle.com/arashnic/hr-analytics-job-change-of-data-scientists?select=aug_train.csv