Introduction to Big Data & Data Analytics

COMPARISON OF K-NN AND DECISION TREE ALGORITHMS

TO

MR. BHARAT GUPTA

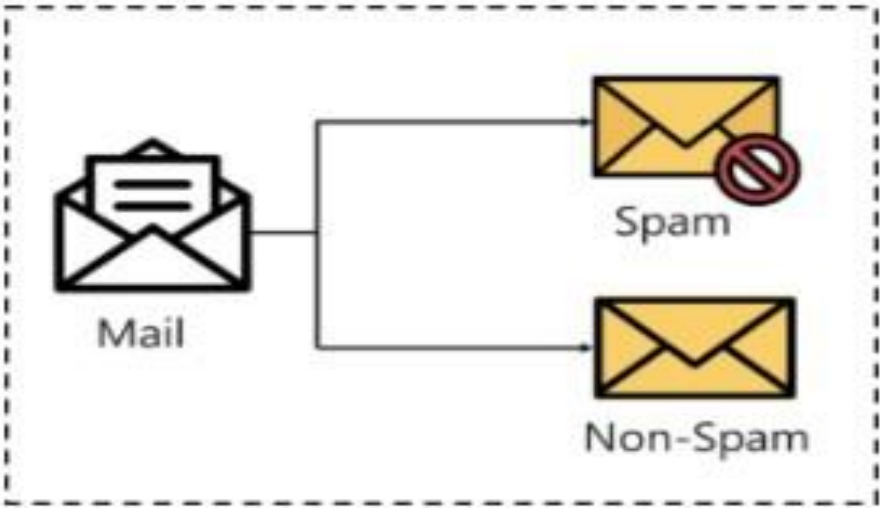
BY

RHEA JAIN 20803018 B-12

ABSTRACT

Classification is a process of categorizing a given set of data into classes, it can be performed on both structured or unstructured data. The main goal is to identify which class/category the new data will fall into.

The most common classification problems are – speech recognition, face detection, handwriting recognition, document classification, etc.



There are a bunch of machine learning algorithms for classification in machine learning. Two of them are **K-Nearest Neighbor** and **Decision Tree Algorithms**

Both, K-NN and decision trees are supervised algorithms. They both require labelled training data in order to label the test data.

TABLE OF CONTENTS

INTRODUCTION

K-NN and **Decision tree** are very popular classifiers. But which one is more accurate?

In this mini project we have done the complete analysis of both the algorithms by calculating the **Jaccard Score** and **F1-score** of both the algorithms using a **Loan Dataset** taken from Kaggle.

We have also done the **Data Visualization** of the dataset.

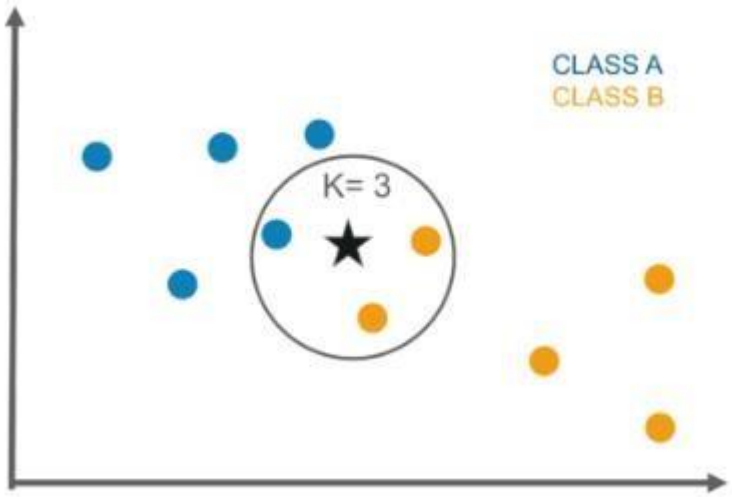
These algorithms help to classify the customer on the basis of loan status i.e., **PAIDOFF** and **COLLECTION**.

We have done the model evaluation by taking a test set from Kaggle only.

Related Work / Literature Survey

1. K-NEAREST NEIGHBOR ALGORITHM

It is a lazy learning algorithm that stores all instances corresponding to training data in n-dimensional space**.** It is a **lazy learning algorithm** as it does not focus on constructing a general internal model, instead, it works on storing instances of training data.



Classification is computed from a simple majority vote of the k nearest neighbors of each point. It is supervised and takes a bunch of labeled points and uses them to label other points. To label a new point, it looks at the labeled points closest to that new point also known as its nearest neighbors. It has those neighbors vote, so whichever label the most of the neighbors have is the label for the new point. The “k” is the number of neighbors it checks.

Advantages of KNN

1. **No Training Period:** KNN is called **Lazy Learner (Instance based learning)**. This makes the KNN algorithm much faster than other algorithms that require training.

2. Since the KNN algorithm requires no training before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm.

3. KNN is very easy to implement. There are only two parameters required to implement KNN i.e. the value of K and the distance function.

Disadvantages of KNN

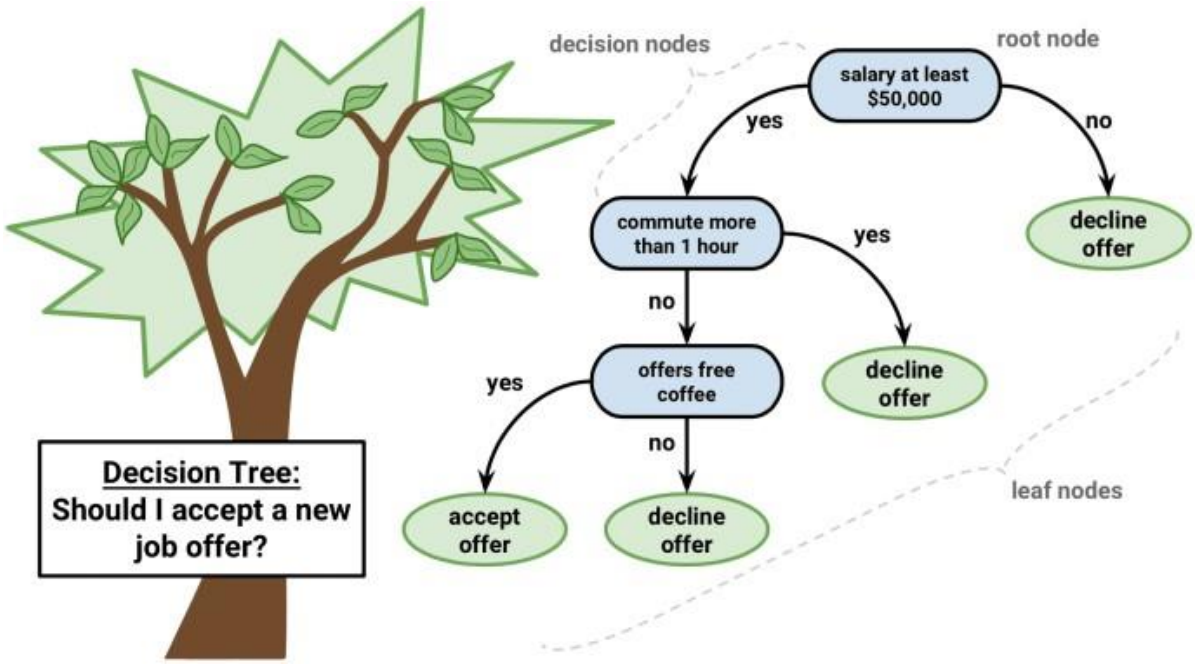
1. **Does not work well with large dataset:** In large datasets, the cost of calculating the distance between the new point and each existing points is huge which degrades the performance of the algorithm.

2. **Need feature scaling:** We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions.

2. DECISION TREE

The decision tree algorithm builds the classification model in the form of a tree structure. It utilizes the if-then rules which are equally exhaustive and mutually exclusive in classification. The process goes on with breaking down the data into smaller structures and eventually associating it with an incremental

decision tree. The final structure looks like a tree with nodes and leaves. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covering the rules are removed. The process continues on the training set until the termination point is met.



The tree is constructed in a top-down recursive divide and conquer approach. A decision node will have two or more branches and a leaf represents a classification or decision. The topmost node in the decision tree that corresponds to the best predictor is called the root node, and the best thing about a decision tree is that it can handle both categorical and numerical data.

ADVANTAGES OF DECISION TREE

1. A decision tree does not require normalization of data.
2. A decision tree does not require scaling of data as well.
3. A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

DISADVANTAGES OF DECISION TREE

1. A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.
2. For a Decision tree sometimes, calculation can go far more complex compared to other algorithms.
3. For a Decision tree sometimes, calculation can go far more complex compared to other algorithms.

TOOLS / TECHNOLOGIES USED

Text editor

Data set collection

kaggle

Language Used

Libraries Used

1. **scikit-learn**
2. **pandas**
3. **matplotlib**
4. **numpy**

DATASET USED

We have taken the Loan data set from Kaggle. This dataset is about past loans. The **Loan_train.csv** data set includes details of 346 customers whose loan are already paid off or defaulted. It includes following fields:

Field	Description
Loan_status	Whether a loan is paid off on in collection
Principal	Basic principal loan amount at the
Terms	Origination terms which can be weekly (7 days), biweekly, and monthly payoff schedule
Effective_date	When the loan got originated and took effects
Due_date	Since it's one-time payoff schedule, each loan has one single due date
Age	Age of applicant
Education	Education of applicant
Gender	The gender of applicant

We load a dataset using Pandas library, and apply the mentioned algorithms, and find the best one for this specific dataset by accuracy evaluation methods.

In [3]:
```
df = pd.read_csv('loan_train.csv')
#default 5 rows
df.head()
```

Out[3]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	education	Gender
0	0	0	PAIDOFF	1000	30	9/8/2016	10/7/2016	45	High School or Below	male
1	2	2	PAIDOFF	1000	30	9/8/2016	10/7/2016	33	Bechalor	female
2	3	3	PAIDOFF	1000	15	9/8/2016	9/22/2016	27	college	male
3	4	4	PAIDOFF	1000	30	9/9/2016	10/8/2016	28	college	female
4	6	6	PAIDOFF	1000	30	9/9/2016	10/8/2016	29	college	male

DATA VISUALISATION

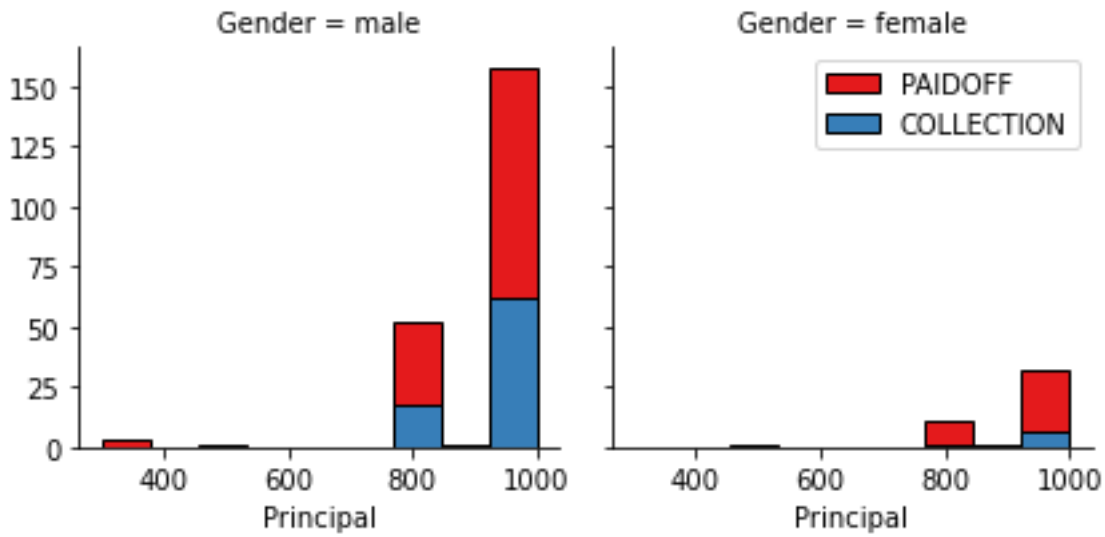


Fig1: principal vs loan status for both the genders

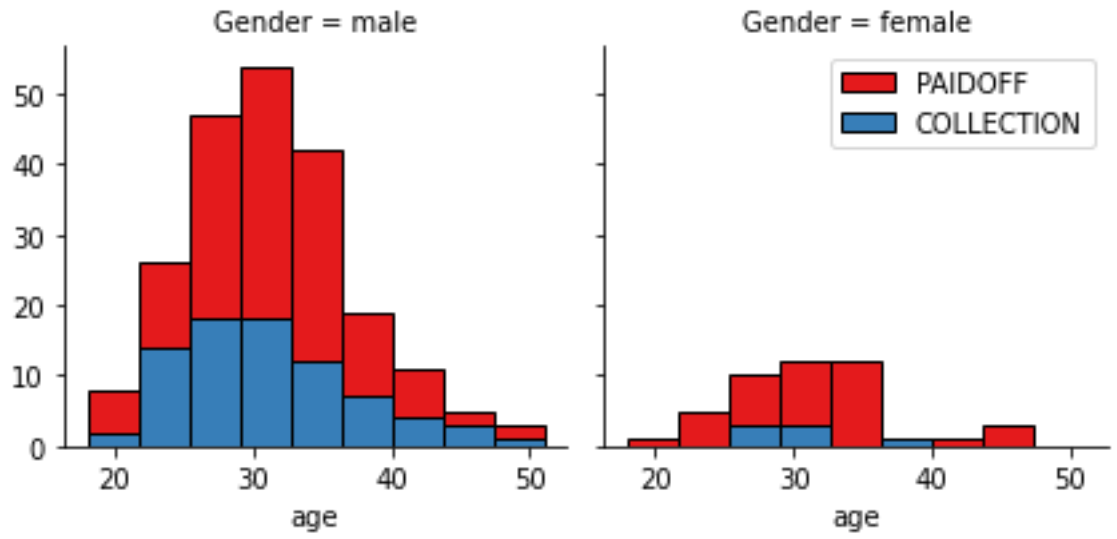


Fig2: age vs loan status for both the genders

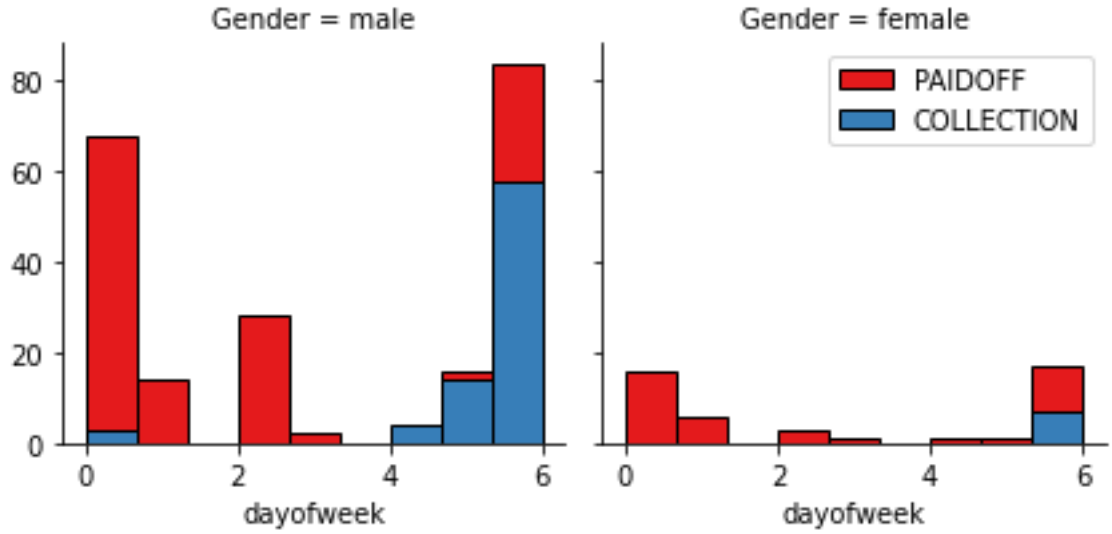


Fig3: Day of week vs loan status for both the genders

Convert Categorical features to numerical values ¶

Lets look at gender:

```
In [12]: df.groupby(['Gender'])['loan_status'].value_counts(normalize=True)

Out[12]: Gender  loan_status
         female  PAIDOFF        0.865385
                 COLLECTION     0.134615
         male    PAIDOFF        0.731293
                 COLLECTION     0.268707
         Name: loan_status, dtype: float64
```

86 % of female pay their loans while only 73 % of males pay their loan

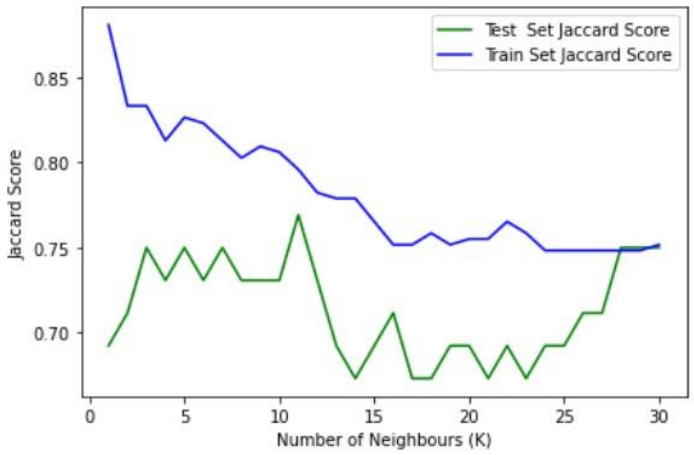
```
In [14]: df.groupby(['education'])['loan_status'].value_counts(normalize=True)

Out[14]: education             loan_status
         Bechalor              PAIDOFF        0.750000
                               COLLECTION     0.250000
         High School or Below  PAIDOFF        0.741722
                               COLLECTION     0.258278
         Master or Above       COLLECTION     0.500000
                               PAIDOFF        0.500000
         college               PAIDOFF        0.765101
                               COLLECTION     0.234899
         Name: loan_status, dtype: float64
```

Fig4: Loan status based on education of people

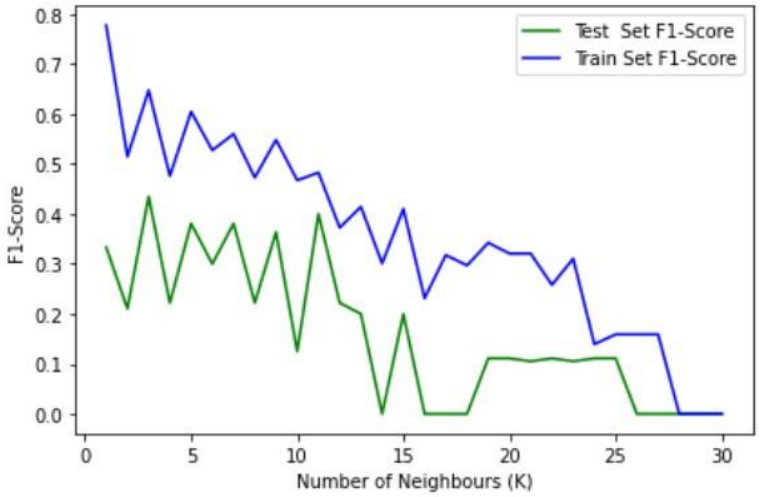
EXPERIMENTAL RESULTS

1. KNN

In [54]:
```
# Plot Jaccard score
plt.plot(range(1, max_k + 1), test_jaccard, 'g', label='Test  Set Jaccard Score')
plt.plot(range(1, max_k + 1), train_jaccard, 'b', label='Train Set Jaccard Score')
plt.legend()
plt.ylabel('Jaccard Score')
plt.xlabel('Number of Neighbours (K)')
plt.tight_layout()
plt.show()
```



In [55]:
```
# Plot F1-score
plt.plot(range(1, max_k + 1), test_f1, 'g', label='Test  Set F1-Score')
plt.plot(range(1, max_k + 1), train_f1, 'b', label='Train Set F1-Score')
plt.legend()
plt.ylabel('F1-Score')
plt.xlabel('Number of Neighbours (K)')
plt.tight_layout()
plt.show()
```



2. Decision Tree

In [58]:
```
# Predict test set
target_pred = dec_tree.predict(features_test)

# Evaluate results - Jaccard score
test_jaccard = metrics.accuracy_score(target_test, target_pred)
train_jaccard = metrics.accuracy_score(target_train, dec_tree.predict(features_train))

# Evaluate results - F1-score
test_f1 = metrics.f1_score(target_test, target_pred)
train_f1 = metrics.f1_score(target_train, dec_tree.predict(features_train))

print(f'Test Jaccard score is {test_jaccard}')
print(f'Test F1-score is {test_f1}')
```

```
Test Jaccard score is 0.6730769230769231
Test F1-score is 0.32
```

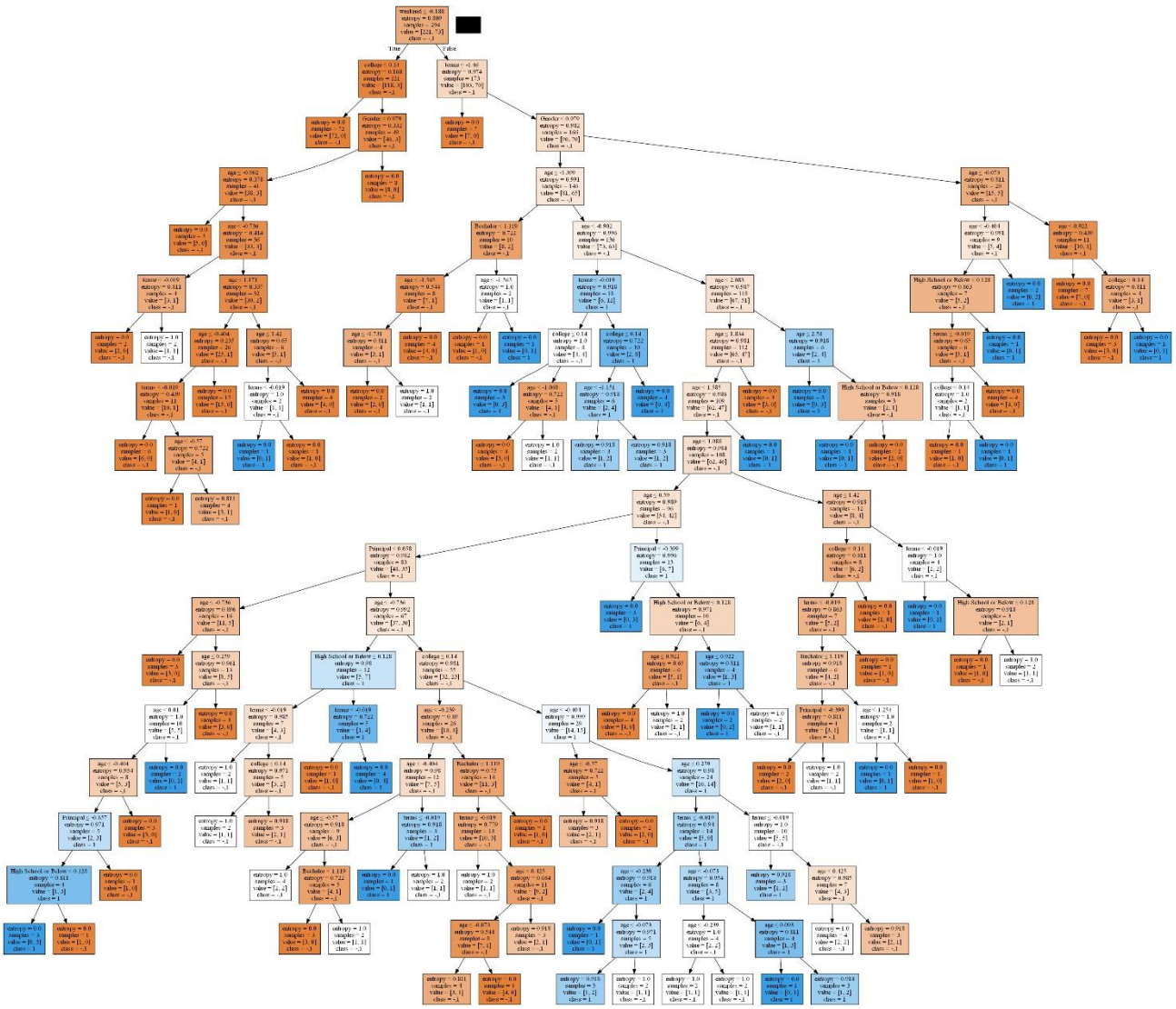


Fig5: Decision tree

Test Bed

Load Test set for evaluation

In [62]:
```
test_df = pd.read_csv('loan_test.csv')
test_df.head()
```

Out[62]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	education	Gender
0	1	1	PAIDOFF	1000	30	9/8/2016	10/7/2016	50	Bechalor	female
1	5	5	PAIDOFF	300	7	9/9/2016	9/15/2016	35	Master or Above	male
2	21	21	PAIDOFF	1000	30	9/10/2016	10/9/2016	43	High School or Below	female
3	24	24	PAIDOFF	1000	30	9/10/2016	10/9/2016	26	college	male
4	35	35	PAIDOFF	800	15	9/11/2016	9/25/2016	29	Bechalor	male

Applying models to test set

KNN

In [64]:
```
# Predict
pred = best_knn.predict(features)

# Evaluate
jaccard = metrics.accuracy_score(target, pred)
f1 = metrics.f1_score(target, pred)

# Print metrics
print(f'Jaccard score is {jaccard}')
print(f'F1-score is {f1}')
```

```
Jaccard score is 0.6296296296296297
F1-score is 0.09090909090909091
```

Decision Tree

In [65]:
```
# Predict
pred = dec_tree.predict(features)

# Evaluate
jaccard = metrics.accuracy_score(target, pred)
f1 = metrics.f1_score(target, pred)

# Print metrics
print(f'Jaccard score is {jaccard}')
print(f'F1-score is {f1}')
```

```
Jaccard score is 0.7592592592592593
F1-score is 0.5185185185185186
```

Algorithm	Jaccard	F1-score
KNN	0.6296	0.0909
Decision Tree	0.7407	0.5

CONCLUSION

So here we can see from the results that Jaccard score and F1-score of Decision Tree is greater than that of K-Nearest Neighbor. So, for this dataset, Decision Tree is a better choice than KNN Classifier.

REFERENCES

Zhang, Shichao, et al. "Learning k for knn classification." *ACM Transactions on Intelligent Systems and Technology (TIST)* 8.3 (2017): 1-19.

Zhang, S., Li, X., Zong, M., Zhu, X., & Cheng, D. (2017). Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(3), 1-19.

Zhang, Shichao, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Debo Cheng. "Learning k for knn classification." *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, no. 3 (2017): 119.

Zhang, S., Li, X., Zong, M., Zhu, X. and Cheng, D., 2017. Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *8*(3), pp.1-19.

Zhang S, Li X, Zong M, Zhu X, Cheng D. Learning k for knn classification. ACM Transactions on Intelligent Systems and Technology (TIST). 2017 Jan 12;8(3):1-9.

Song, Yan-Yan, and L. U. Ying. "Decision tree methods: applications for classification and prediction." *Shanghai archives of psychiatry* 27.2 (2015): 130.

Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130.

Song, Yan-Yan, and L. U. Ying. "Decision tree methods: applications for classification and prediction." *Shanghai archives of psychiatry* 27, no. 2 (2015): 130.

Song, Y.Y. and Ying, L.U., 2015. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), p.130.

Song YY, Ying LU. Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry. 2015 Apr 25;27(2):130.