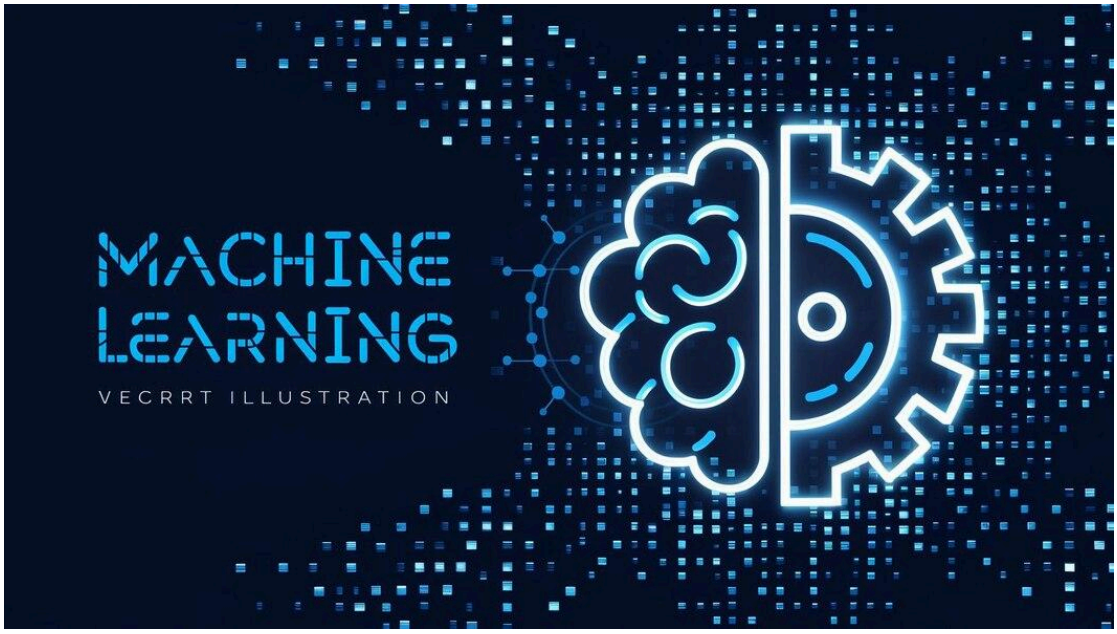

Figure 1. "Machine learning" image is taken from freepik.com.

Data Mining & Machine Learning Project Report

15.11.2024

Ivan Lapickij

4th-year student in
Software Design with Artificial Intelligence in Cloud at
TUS(Technological University of Shanon) Athlone, Ireland

Overview

This is my fourth year of Data mining and machine learning project, for which I chose two datasets: “Abalone” and “Employees”. Then I utilized them using linear regression and decision Tree techniques to predict target variables and test the quality of the prediction. I will also present a workflow and documentation supported by the CrispDM data mining process, including images and references.

1. Linear Regression

Abalone dataset

Figure 2. The “Abalone” image is taken from Food Republic.

1.1. Business Understanding/Objective

Using the Abalone dataset, predict the age of Abalone based on the other features. The age is calculated by multiplying the Rings by 1.5. The Rings will serve as the target variable. By estimating age, fisheries can predict optimal harvesting time and preserve younger populations. As well, it could be useful to monitor the health of the abalone population in the ecosystem.

1.2. Data Understanding/Data Exploration

Data is mostly numeric except for Sex features and contains 4176 rows excluding the header row as we can see in “figure 3”.

Read in the Data

```
[79]: df = pd.read_csv("data/abalone_headers.csv")
      print(df.shape)
      df.head()
```

```
(4177, 9)
```

[79]:

	SEX	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

Figure 3. Few rows of actual data from the abalone dataset.

As we can see in “Figure 4” zero values for the field “Height” are being handled which means data don't have missing values except those already handled. Other fields “min” row looks goods no zero values that means no handling is needed.

```
[81]: # remove the row with Height = 0 using a row condition
      df = df[df.Height != 0]
      print(df.shape)
      print(df.describe())
```

```
(4175, 9)
             Length    Diameter       Height  Whole_weight  Shucked_weight  \
count  4175.000000  4175.00000  4175.000000   4175.000000     4175.000000
mean      0.524065     0.40794     0.139583      0.829005        0.359476
std       0.120069     0.09922     0.041725      0.490349        0.221954
min       0.075000     0.05500     0.010000      0.002000        0.001000
25%       0.450000     0.35000     0.115000      0.442250        0.186250
50%       0.545000     0.42500     0.140000      0.800000        0.336000
75%       0.615000     0.48000     0.165000      1.153500        0.502000
max       0.815000     0.65000     1.130000      2.825500        1.488000

       Viscera_weight  Shell_weight        Rings
count     4175.000000   4175.000000  4175.000000
mean         0.180653      0.238834     9.935090
std          0.109605      0.139212     3.224227
min          0.000500      0.001500     1.000000
25%          0.093500      0.130000     8.000000
50%          0.171000      0.234000     9.000000
75%          0.253000      0.328750    11.000000
max          0.760000      1.005000    29.000000
```

Figure 4. Calculations(min, max,avg) based on the abalone dataset.

1.3. Modeling

Moderate correlation between Rings and features. Expect a moderate model as in “Figure 5”.

Distinguishing age groups in the graph is challenging due to the dense clustering in the middle, with no clear boundaries or separation lines representing distinct ages. A quadratic relationship may exist between age and certain features, indicating different populations of abalone across age groups. This dispersion suggests we might achieve moderate prediction accuracy, with better predictions for middle-aged and older abalones, while middle-aged groups remain harder to separate and predict accurately.

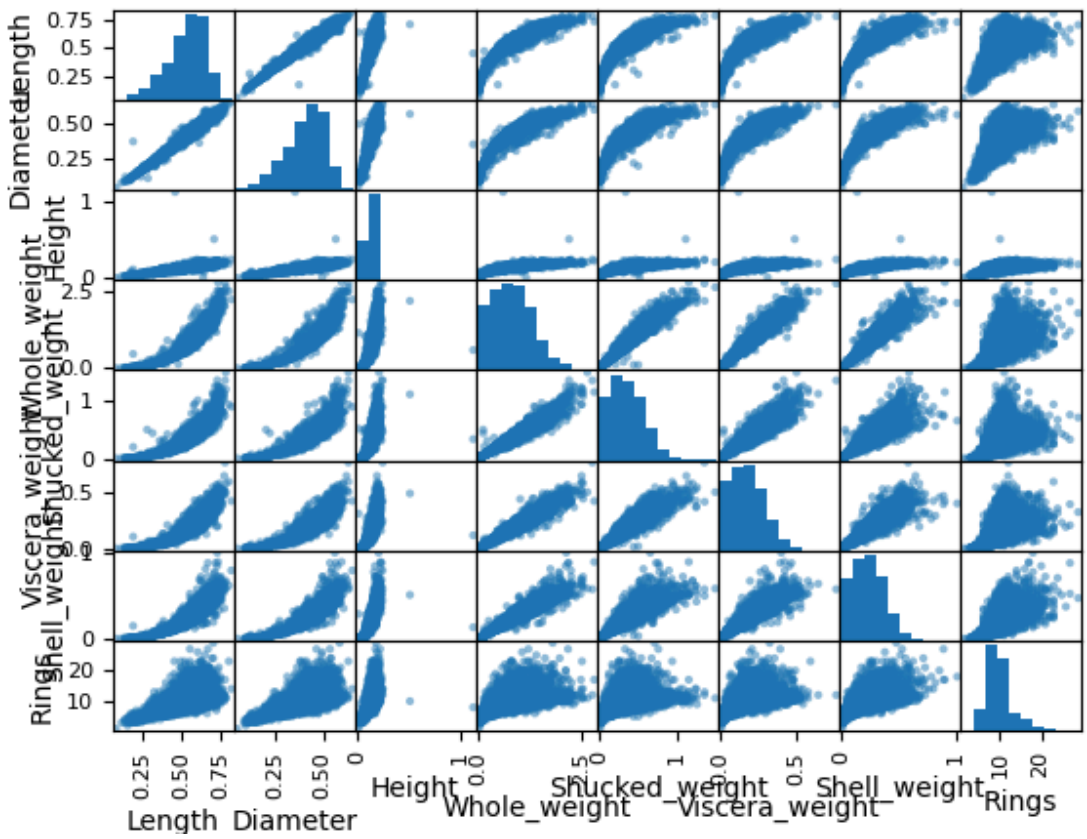

Figure 5. Scatter matrix of abalone dataset.

1.4. Evaluation

As we can see in “Figure 6” we dropped two columns: “Rings” (target variable) and an extra column “Sex” to prove that adding extra features minimizes error & increases the prediction quality indicator.

Prepare the Data

```
X = df.drop(columns=['Rings','SEX'])
y = df.Rings
print(X.shape)
```

```
(4175, 7)
```

```
# split into training and test data
# random_state=1 ensures the split is the same every time we run it

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
# X_train.head()
```

```
(3131, 7)
(3131,)
(1044, 7)
(1044,)
```

Figure 6. Showing workflow of dropping columns and splitting data.

The Coefficient of Determination is equal to 53%, which indicates that 53% of the variance in the target variable is explained by the model, leaving 47% of the variability unaccounted for. This suggests a moderate fit, where the model captures some patterns but may not fully explain the relationship between features and the target. As shown in “Figure 7”.

Build the model ¶

```
model = LinearRegression()
model.fit(X_train, y_train)
print('intercept:', model.intercept_)
print('slope:', model.coef_)
print('Coefficient of Determination R^2 :',  model.score(X_train, y_train))
```

```
intercept: 3.042912983425272
slope: [ -2.0160073   14.2258051    9.9676144    8.92912367 -19.73273439
  -8.83815136   8.38328195]
Coefficient of Determination R^2 : 0.5283630582063565
```

Figure 7. Showing model building and coverage of model(Coefficient of Determination).

RMSE - is an indicator of error range in this case its 2.3 rings, average of rings is about 10.

Claculate the RMSE for the test data

```
yhat = model.predict(X_test)
# print (yhat)
print(mean_squared_error(y_test, yhat, squared=False))
```

```
2.251371123188242
```

**RMSE value 2.23 is reasonably low compared to an average Rings of about 10 **

Figure 8. Indicator of error range.

As we can see in "Figure 9" the feature "Sex" was added to prove to add the feature increases the quality of prediction and reduces the risk of error, since it is not a numeric type I used one hot encoded and got "dummies" that replaced "String" into "Boolean".

Iterate, include SEX as a Feature

```
X = df.drop(columns = 'Rings')
y = df.Rings
X.head()
```

	SEX	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055

```
# one hot encode the Species variable.
X = pd.get_dummies(X)
X.tail()
```

	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	SEX
4172	0.565	0.450	0.165	0.8870	0.3700	0.2390	0.2490	Tr
4173	0.590	0.440	0.135	0.9660	0.4390	0.2145	0.2605	Fa
4174	0.600	0.475	0.205	1.1760	0.5255	0.2875	0.3080	Fa
4175	0.625	0.485	0.150	1.0945	0.5310	0.2610	0.2960	Tr
4176	0.710	0.555	0.195	1.9485	0.9455	0.3765	0.4950	Fa

Figure 9. Dataset table after dropping Rings and adding "hot encode".

1.5. Conclusion

- Adding extra features increases the quality of prediction as well as reduces the size of errors(RMSE).
- RMSE - is low 2.3-2.2 compared to an average of rings 10.
- Linear regression explains half of the current dataset.
- Based on “Figure 10” most of Abalone data regarding age is middle age from 7 to 19 years old from actual 0 min and 44 max that Linear Regression captured.

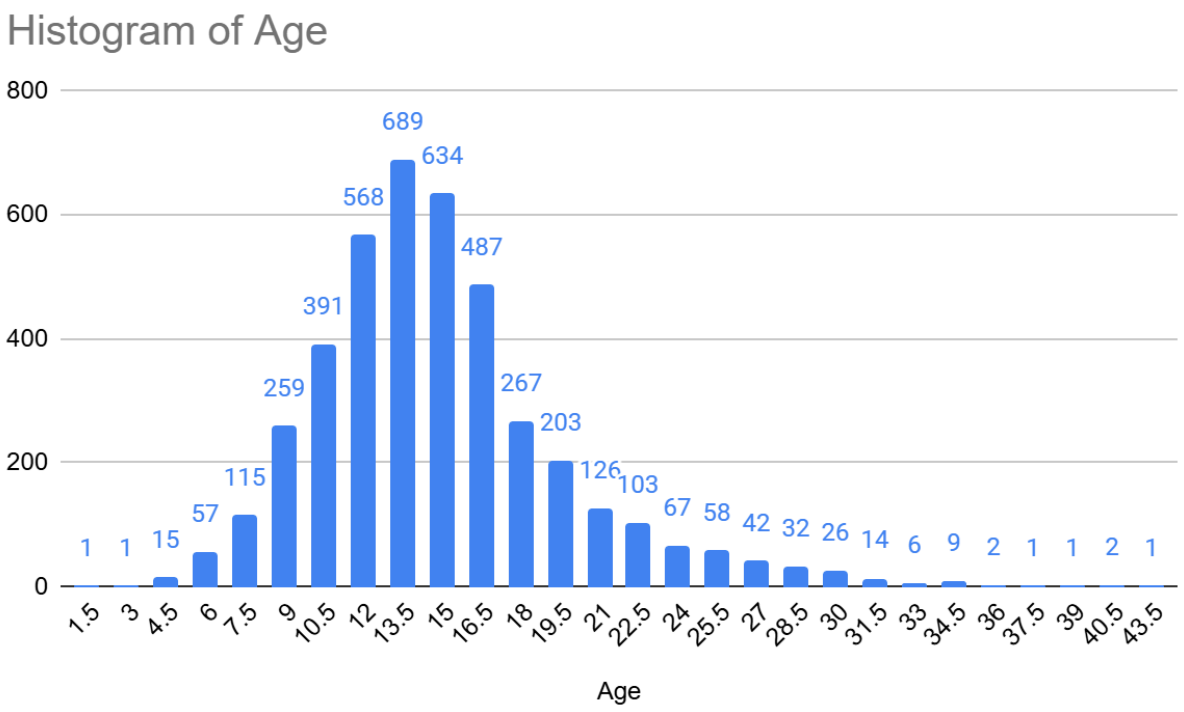

Figure 10. Histogram of Age from CSV data where “Rings” column was used to create “Age” column and create Histogram.

2. Decision tree

Employees dataset

Figure 11. Employees leaving. Image taken from Canada.com.

2.1. Business Understanding/Objective

Predict factors that make employees more likely to leave a company based on various features. This analysis could help identify reasons employees are more inclined to leave or stay, supporting efforts to improve retention and encourage longer employee tenure.

2.2. Data Understanding/Data Exploration

Data have some numeric and sorting types. The total is 4653 Rows of data with 9 columns. As we see at the moment there is several employees to leave as “0” or stay as “1”. No data values missing as the dataset source suggests and from min values where 0 is not a valid value and stands for missing values.

	JoiningYear	PaymentTier	Age	ExperienceInCurrentDomain	LeaveOrNot
count	4653.000000	4653.000000	4653.000000	4653.000000	4653.000000
mean	2015.062970	2.698259	29.393295	2.905652	0.343864
std	1.863377	0.561435	4.826087	1.558240	0.475047
min	2012.000000	1.000000	22.000000	0.000000	0.000000

Figure 12. Employees dataset Calculations of overall features table.

Target column "LeaverOrNo" is dropped and a test size of 1000 is chosen for the best training result and accuracy.

Build the Model

```
# Encode categorical data
df_encoded = pd.get_dummies(df, drop_first=True)
# Separate features and target
X = df_encoded.drop(columns='LeaveOrNot')  # Adjust based on your actual target column nam
y = df_encoded['LeaveOrNot']
```

```
# 125 training and 25 test
# stratify ensures the same proportions of species in traing and test data.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1000, random_state=1,
```

Figure 13. Showing model is being built.

2.3. Modeling

As in "Figure 14" the grouping is obvious and corresponds to different features. It seems to split with gaps and lines. Separating employees who leave and stay should be easy. PaymentTier & ExperienceInCurrentDomain looks to be promising features.

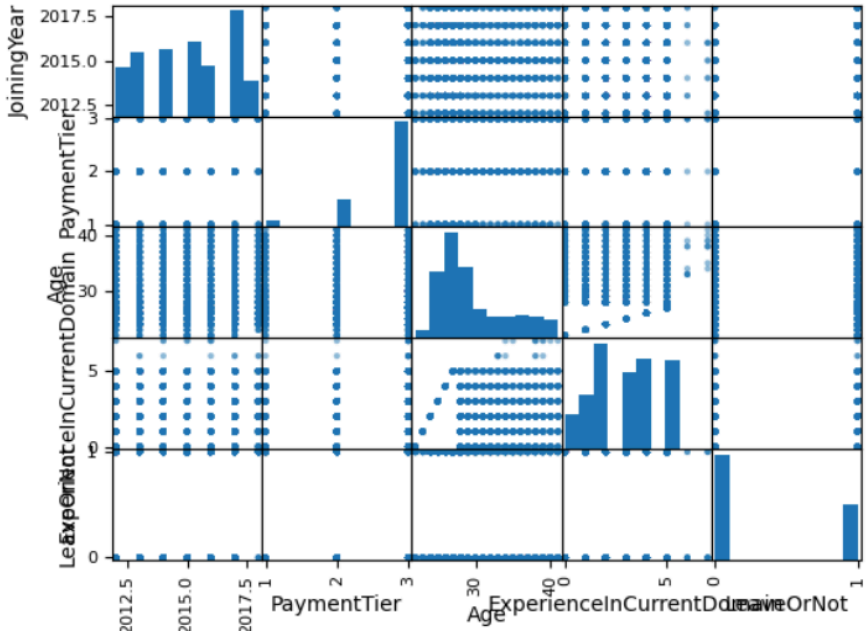

Figure 14. Scatter matrix of employees dataset.

The plot in "Figure 15" shows that the most recent employees are most likely to leave, disregarding the age group.

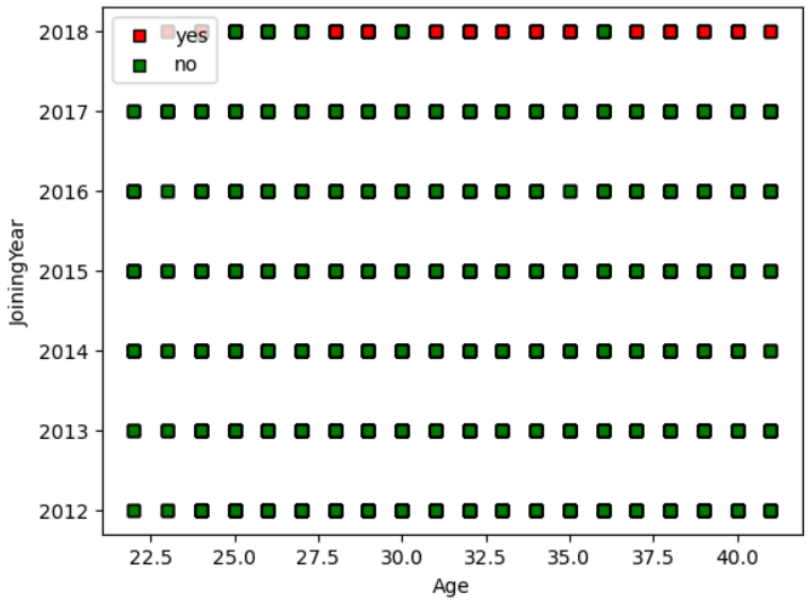

Figure 15. The plot shows the relationship between Age and joining year.

The plot in "Figure 15" shows that the youngest employees with the lowest PaymentTier are the ones most likely to leave.

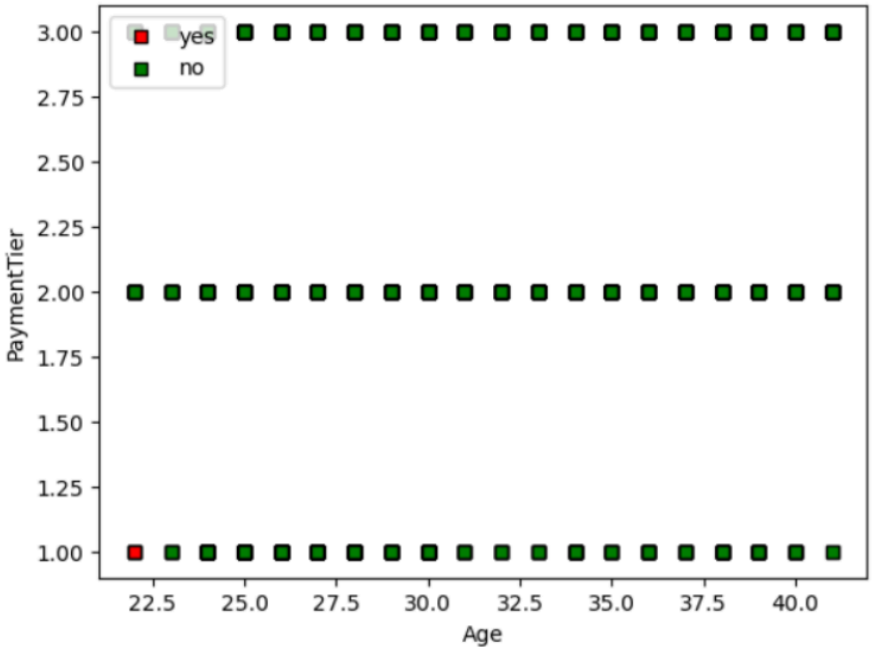

Figure 15. The plot shows the relationship between PaymentTier and employers' Age.

.

2.4. Evaluation

As we can see we are getting a pretty good model with high training and test accuracy, “Figure 16”. This indicates model has a pretty good prediction that covers most of the data.

Evaluation ¶

```
print("[training accuracy: ", model.score(X_train, y_train), "]")
print("test accuracy: ", model.score(X_test, y_test))
```

```
[training accuracy:  0.9315630988228853 ]
test accuracy:  0.827
```

Figure 16. Model evaluation.

Further model is going as far as a depth of 23. This means the longest branch of the decision tree contains 23 nodes from the root branch to the leaf node.

Figure 17. Long decision tree.

Since the Long decision tree is too long and had to be scaled, I narrowed the depth to 3 to see what's happening and how Gini impurity influences the decision paths by splitting value to the leaf node.

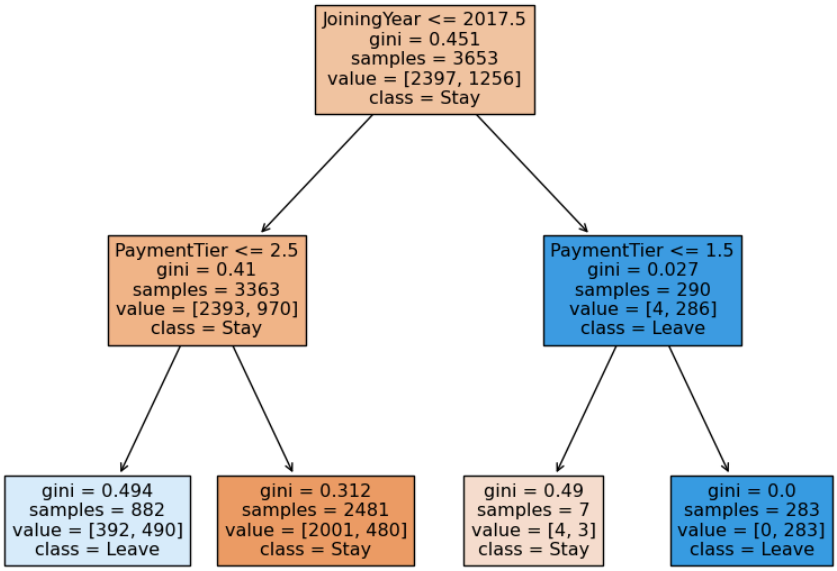

Figure 17. Long decision tree.

2.5. Conclusion

- As we could notice from the evaluation model is pretty successful in prediction and we got a huge decision tree with a long path of 23 nodes till the leaf.
- Doesn’t seem like a tree overfit because the gap between training and test data wasn't huge.
- The tree could be pruned reducing depth or choosing minimum split functions to overcome overfit or complex decision trees.
- The model could be used in the real system to evaluate factors that encourage employees to stay.

Figure 17. Diagram of Employees leaving or staying decisions based on CSV Excel spreadsheet - employees dataset.

References

[Employee dataset]

[Abalone - UCI Machine Learning Repository]

https://github.com/IvanLapickij/DMML_Assesment