Project Topic :

EMPLOYEE RETENTION SYSTEM USING DATA SCIENCE, ML AND DATA ANALYTICS

Predicting which employee will probably leave the company.

Group Members:
Faiz Patel
Sourav Sajeevan
Akash Pansambal

Under the guidance:
Assistant Prof. Miss Shital Jadhav

Contents

Model Training

1. Decision Tree
2. KNN

Accuracy of models

Predicting and Custom Data

Accuracy Comparison

Literature survey

Abstract

For organizations to thrive in today’s economy, finding and retaining the best employees is vital. However, offering a high salary isn’t the only way to compete with larger employers benefits play a large role in employee retention. Therefore, we are going to build a software that will predict whether an employee will stay for a long time in a company or he is more likely to leave the company so that organization can take initiatives in advance and provide him/her necessities in order to increase his/her satisfaction.

Introduction

Many of the employees leave the company with no clear reasons which causes certain problem. Retention of employees within an organization has become an important issue as it has become difficult to find out why employees are leaving an organization and to keep them satisfied is a big challenge. In this project we are going to use data set which contains employees' data with important attributes. After cleaning and analyzing the data, we are going to use decision tree algorithm and K Nearest Neighbor (KNN) algorithm simultaneously. So that we can know which way is more efficient to predict the outcomes.

Problem Statement

Let’s see the necessity of the initiative we are taking.

Current Scenario

Currently, funded technology start-ups are seeing the highest level of attrition at 30% whereas IT product firms are seeing only half of that. Domestic IT services providers and large consulting firms are witnessing a 25% attrition while GICs/captives and non-IT enterprises are reporting an overall tech talent attrition of 20%, according to data from Xpheno.

- 21% Tech Mahindra Employees Resigned In 90 Days: Attrition Rate Double Of TCS!
  -Track.in

- Cognizant faces the worst attrition rate in the industry, hires record number of employees to fill the gap.
  -Businessinsider.in

A glance over situation

A GLANCE OVER DATA SET

Table 1:

```
1 hr_df
```

	employee_id	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	department	salary
0	1003	2	157	3	0	1	0	sales	low
1	1005	5	262	6	0	1	0	sales	medium
2	1486	7	272	4	0	1	0	sales	medium
3	1038	5	223	5	0	1	0	sales	low
4	1057	2	159	3	0	1	0	sales	low
...	...	...	...	...	...	...	...	...	...
14994	87670	2	151	3	0	1	0	support	low
14995	87673	2	160	3	0	1	0	support	low
14996	87679	2	143	3	0	1	0	support	low
14997	87681	6	280	4	0	1	0	support	low
14998	87684	2	158	3	0	1	0	support	low

14999 rows × 9 columns

Table 2:

```
1 s_df
```

	EMPLOYEE #	satisfaction_level	last_evaluation
0	1003	0.38	0.53
1	1005	0.80	0.86
2	1486	0.11	0.88
3	1038	0.72	0.87
4	1057	0.37	0.52
...	...	...	...
14994	87670	0.40	0.57
14995	87673	0.37	0.48
14996	87679	0.37	0.53
14997	87681	0.11	0.96
14998	87684	0.37	0.52

14999 rows × 3 columns

Checking and Filling Null Values

Filling null values by mean of the columns

```
1  main_df.fillna(main_df.mean(), inplace = True)
```

```
1  main_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 11 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   employee_id            14999 non-null  int64
 1   number_project         14999 non-null  int64
 2   average_montly_hours   14999 non-null  int64
 3   time_spend_company     14999 non-null  int64
 4   Work_accident          14999 non-null  int64
 5   left                   14999 non-null  int64
 6   promotion_last_5years  14999 non-null  int64
 7   department             14999 non-null  object
 8   salary                 14999 non-null  object
 9   satisfaction_level     14999 non-null  float64
 10  last_evaluation        14999 non-null  float64
dtypes: float64(2), int64(7), object(2)
memory usage: 1.3+ MB
```

Libraries and Algorithms:

1. **Numpy**
   For numerical operations on data set.

2. **Pandas**
   To handle Null values and manipulating data set.

3. **Matplotlib / Seaborn**
   For Data visualization

4. **Decision Tree**

5. **KNN**

Numerical Analysis

Value Counts of DEPARTMENT & LEFT column

```
In [19]:  1  main_df['department'].value_counts()
```

```
Out[19]: sales          4140
         technical      2720
         support        2229
         IT             1227
         product_mng     902
         marketing       858
         RandD           787
         accounting      767
         hr              739
         management      630
         Name: department, dtype: int64
```

In [21]:
```
1 main_df.groupby('department').sum()
```

Out[21]:

department	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	satisfaction_level	last_evaluation
IT	4683	248119	4256	164	273	3	758.17283	879.452250
RandD	3033	158030	2650	134	121	27	487.80000	560.446125
accounting	2934	154292	2702	96	204	14	446.68283	550.706125
hr	2701	146828	2480	89	215	15	442.53566	524.006125
management	2432	126787	2711	103	91	69	391.76566	456.234499
marketing	3164	171073	3063	138	203	43	530.62283	613.946125
product_mng	3434	180369	3135	132	198	0	559.19566	644.662250
sales	15634	831773	14631	587	1014	100	2543.77981	2938.236749
support	8479	447490	7563	345	555	20	1377.90849	1611.534499
technical	10548	550793	9279	381	697	28	1653.37264	1961.930624

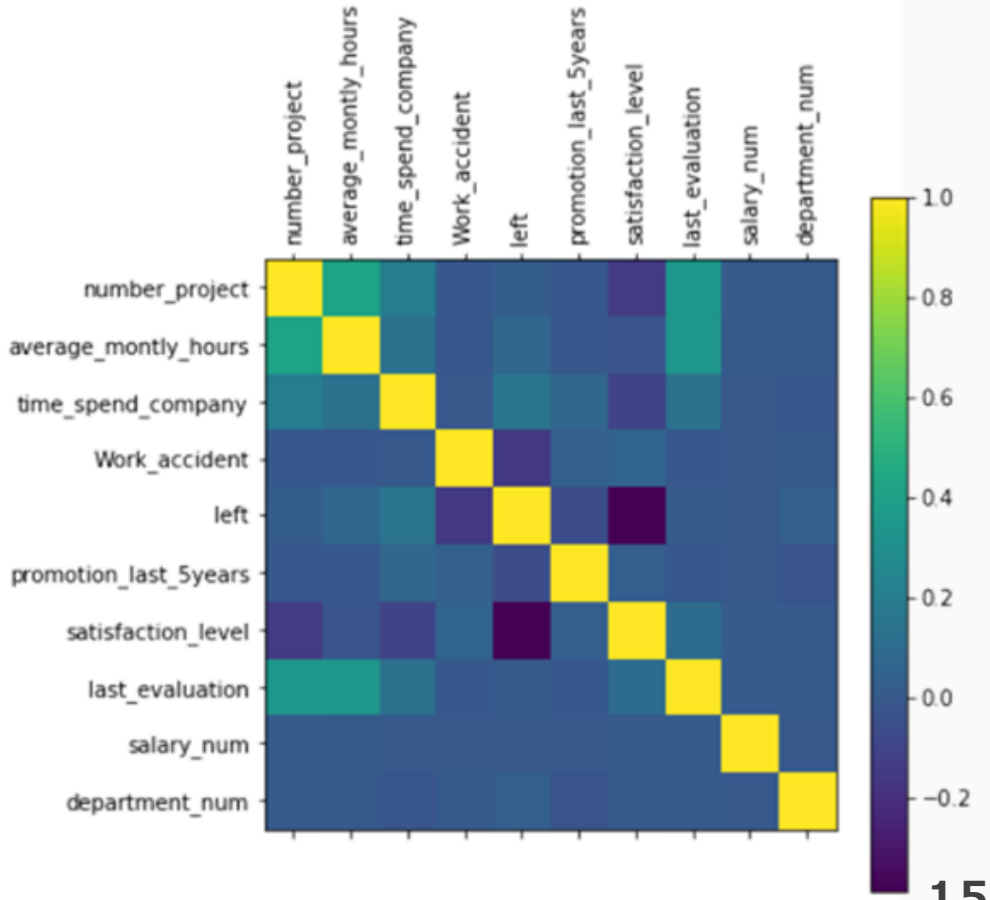
In [22]:
```
1 main_df.groupby('department').mean()
```

Out[22]:

department	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	satisfaction_level	last_evaluation
IT	3.816626	202.215974	3.468623	0.133659	0.222494	0.002445	0.617908	0.716750
RandD	3.853875	200.800508	3.367217	0.170267	0.153748	0.034307	0.619822	0.712130
accounting	3.825293	201.162973	3.522816	0.125163	0.265971	0.018253	0.582377	0.718000
hr	3.654939	198.684709	3.355886	0.120433	0.290934	0.020298	0.598830	0.709075
management	3.860317	201.249206	4.303175	0.163492	0.144444	0.109524	0.621850	0.724182
marketing	3.687646	199.385781	3.569930	0.160839	0.236597	0.050117	0.618442	0.715555
product_mng	3.807095	199.965632	3.475610	0.146341	0.219512	0.000000	0.619951	0.714703
sales	3.776329	200.911353	3.534058	0.141787	0.244928	0.024155	0.614440	0.709719
support	3.803948	200.758188	3.393001	0.154778	0.248991	0.008973	0.618173	0.722985
technical	3.877941	202.497426	3.411397	0.140074	0.256250	0.010294	0.607858	0.721298

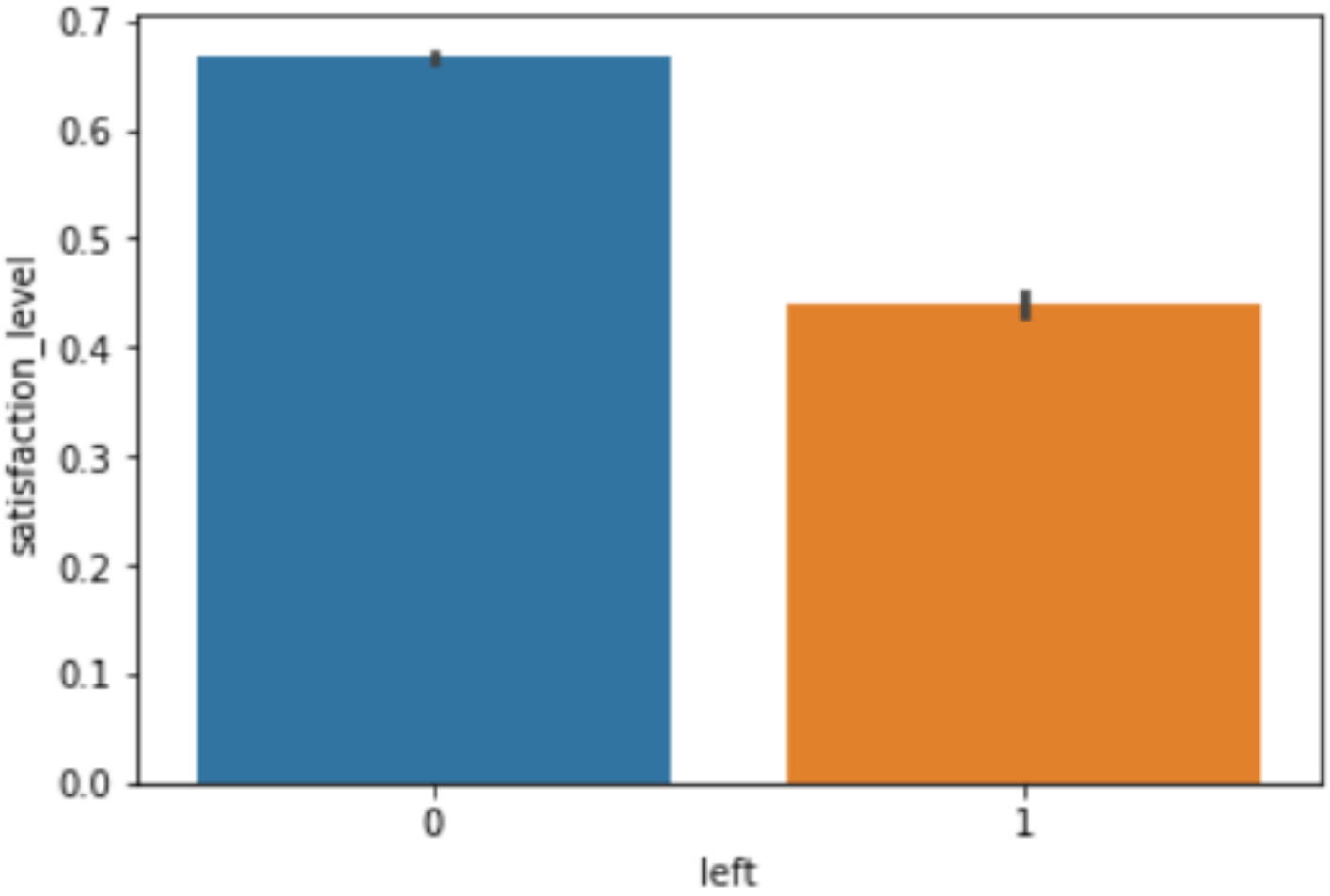
Graphical Analysis

```
1  def plot_cor(df, size = 6):
2      corr = df.corr()
3      fig, ax = plt.subplots(figsize = (size, size))
4  #      ax.legend()
5      cax = ax.matshow(corr)
6      fig.colorbar(cax)
7      plt.xticks(range(len(corr.columns)), corr.columns, rotation = 'vertical')
8      plt.yticks(range(len(corr.columns)), corr.columns)
9
10 plot_cor(main_df)
```



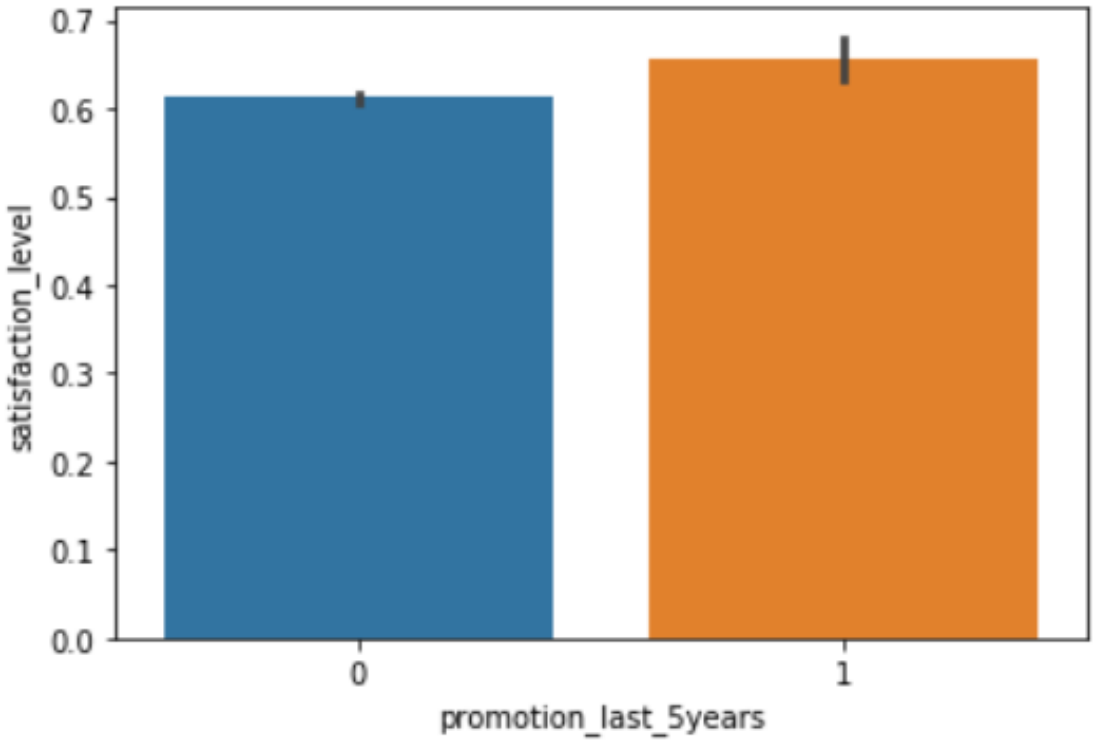
```
1 sns.barplot(x = 'left', y = 'satisfaction_level', data = main_df)
```

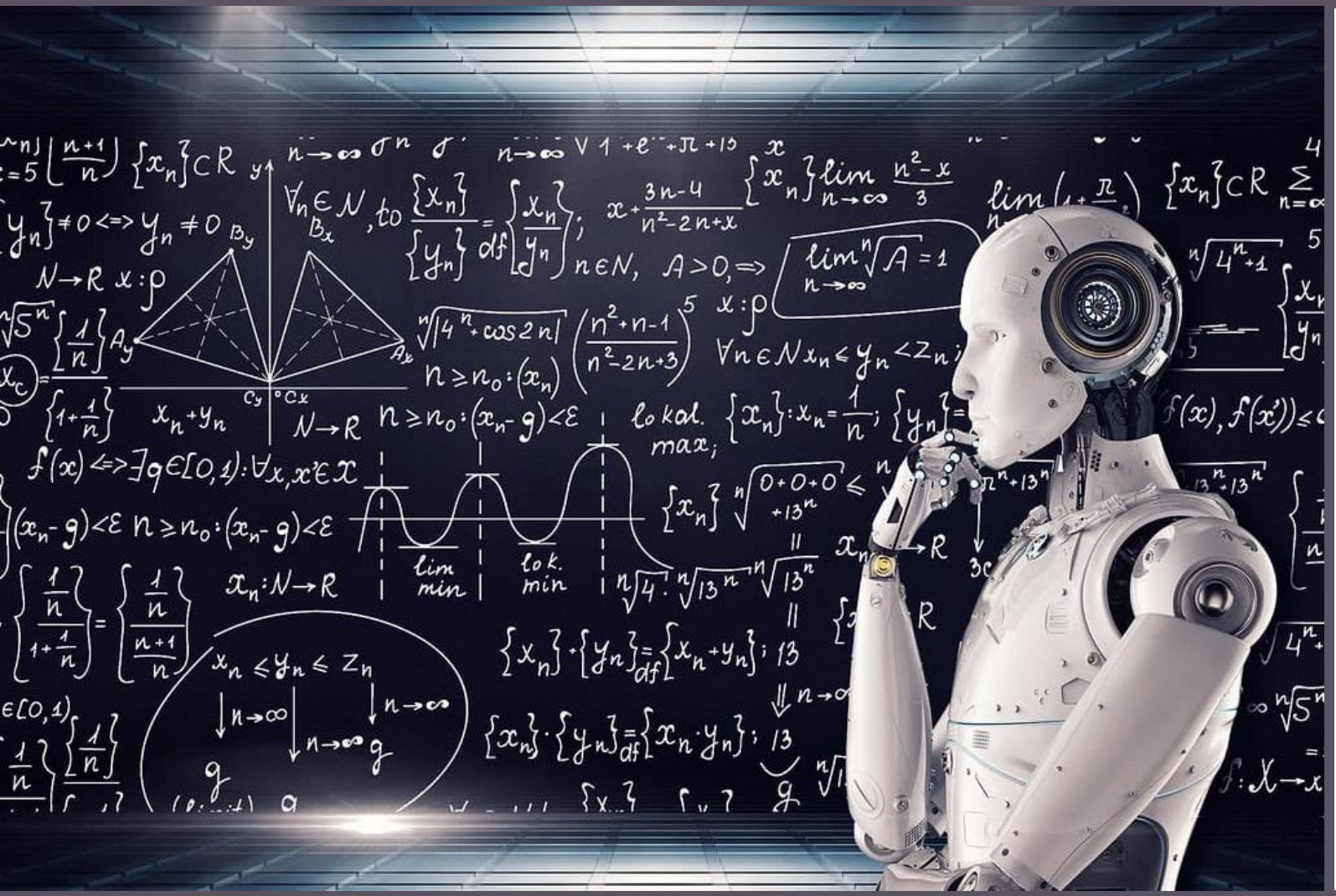
```
<matplotlib.axes._subplots.AxesSubplot at 0x26379f105c8>
```



In [48]:
```
1 sns.barplot(x = 'promotion_last_5years', y = 'satisfaction_level', data = main_df)
```

Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x2631c306848>





Training Models

We used 2 algorithms.

1. Decision Tree Algorithm
2. KNN Algorithm

1. Decision Tree Model Training

In [87]:
```
1 from sklearn.tree import DecisionTreeClassifier
```

In [89]:
```
1 dt = DecisionTreeClassifier()
2 dt.fit(x_train, y_train)
```

Out[89]:
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

In [90]:
```
1 prediction_dt = dt.predict(x_test)
```

2. KNN Classifier

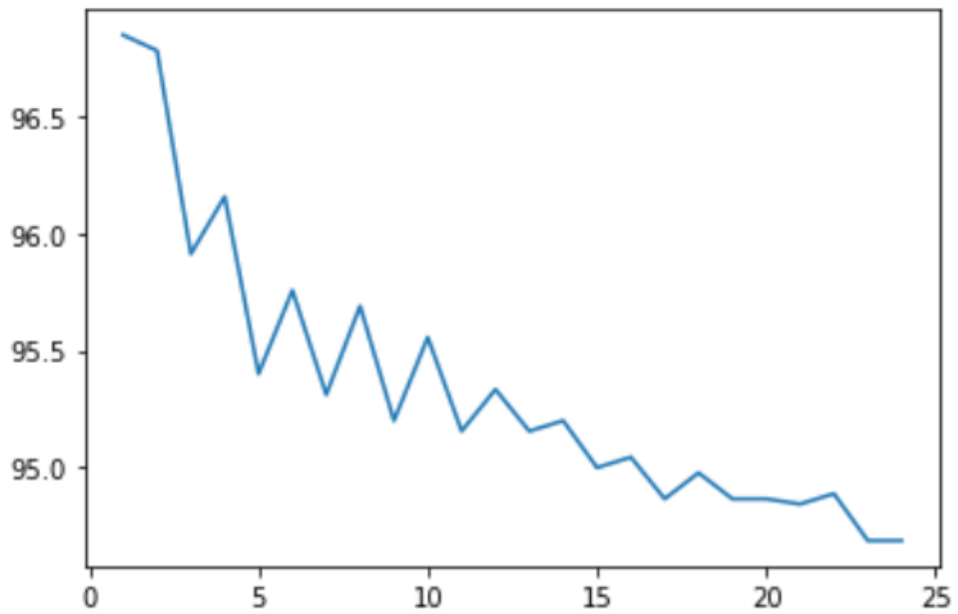
Finding correct value for K

```
scores = {}
score_list = []

for i in range(1, 25):
    knnz = KNeighborsClassifier(n_neighbors = i)
    knnz.fit(x_train_std, y_train)
    predicted_knn = knnz.predict(x_test_std)
    scores[i] = accuracy_score(predicted_knn, y_test)*100
    score_list.append(scores[i])
```

```
plt.plot(range(1, 25), score_list)
```

```
[<matplotlib.lines.Line2D at 0x2631edf2688>]
```



In [174]:
```
1 from sklearn.neighbors import KNeighborsClassifier
2
3 knn = KNeighborsClassifier(n_neighbors = 4)
4 knn.fit(x_train_std, y_train)
```

Out[174]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=4, p=2,
                     weights='uniform')
```

In [175]:
```
1 predicted_knn = knn.predict(x_test_std)
```

Accuracy of models

Accuracy of Decision Tree Classifier

```
In [100]:
1 from sklearn.metrics import accuracy_score

In [185]:
1 accuracy_dt = accuracy_score(y_test, prediction_dt)*100

In [186]:
1 accuracy_dt

Out[186]: 97.4888888888889
```

Accuracy of KNN

```
In [219]:
1 accuracy_knn = accuracy_score(predicted_knn, y_test) *100
2 accuracy_knn

Out[219]: 96.15555555555557
```

Predicting on Custom Data

In [ ]:
```
1 category = ['Employee will stay...', 'Employee will leave...']
```

In [182]:
```
1 data = np.array([10, 135, 5, 0, 1, 0.78, 0.96, 2, 8]).reshape(1, -1)
2 data_std = sc.transform(data)
```

From Decision Tree

In [183]:
```
1 print(category[int(dt.predict(data))])
```

```
Employee will stay...
```

From KNN

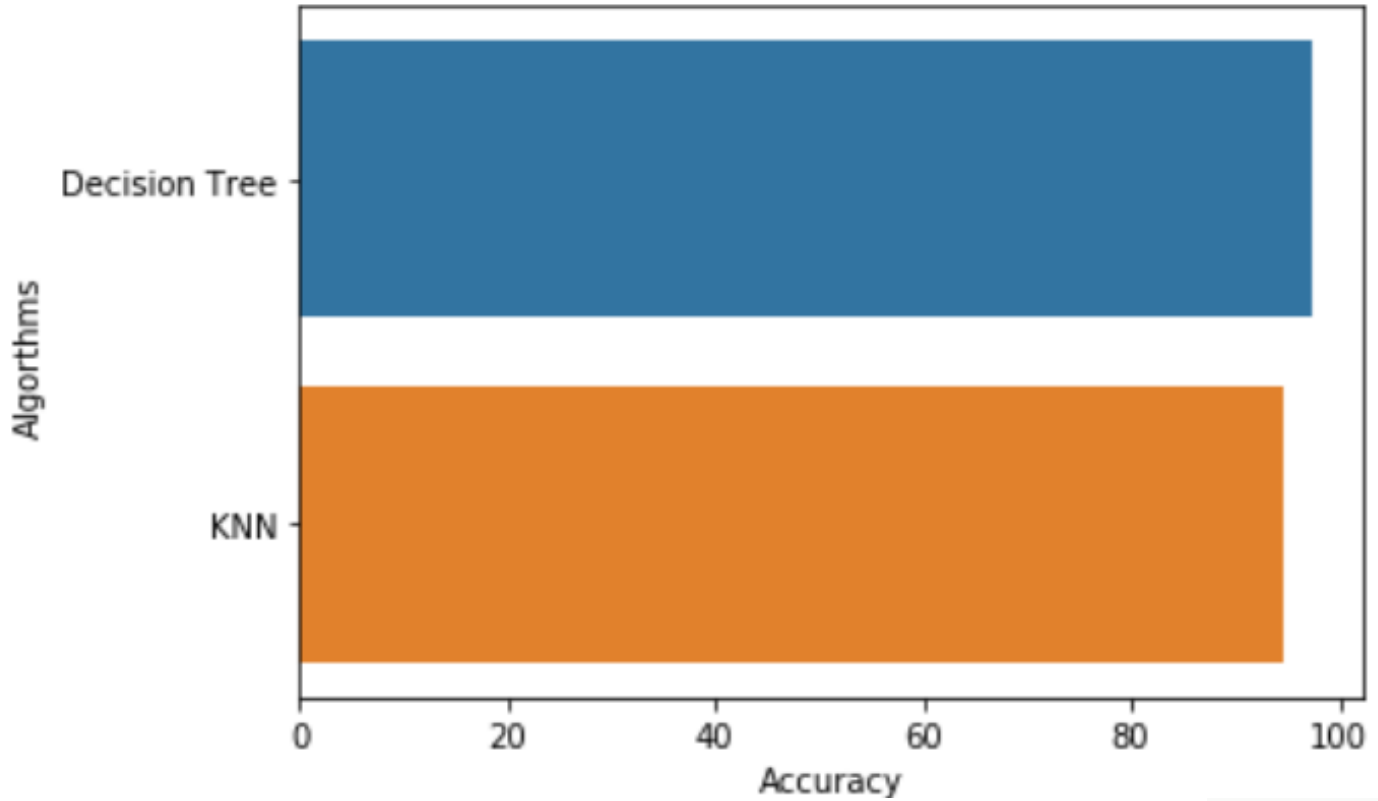
In [184]:
```
1 print(category[int(knn.predict(data_std))])
```

```
Employee will stay...
```

Accuracy Comparison of models

In [208]:
```
algorithms = ['Decision Tree', 'KNN']
scores = [accuracy_dt, accuracy_knn]
```

In [211]:
```
plt.xlabel("Accuracy")
plt.ylabel("Algorthms")
sns.barplot(scores, algorithms)
plt.show()
```



Literature Survey

So many efforts were made to find proper employee management in companies, we are discussing some of the work from them.

1. “Le Zhang” and “Graham Williams” proposed that employee retention is the biggest challenge for a company. They used R for predictions by feature extraction methods a word-to-vector, term frequency and inverse document frequency, R packages such a tm etc. They finally concluded that ensemble techniques can be deployed to effectively boost model performance.

2. “Rupesh Khare”, “Dimple Kaloya” and “Gauri Gupta” proposed that a risk equation can be develop, which can be used assess attrition risk with current set of employees that a company is having. They concluded by stating that among the various attrition predictive techniques available in the market, Logistic Regression and Discriminant Analysis are the closest to give a solution which produced highly accurate results.

THANK YOU