FINAL PROJECT 3: WHY EMPLOYEES LEAVE

Raymond Delacruz

TOPIC/PROBLEM

- This project analyzes a simulated data set on why experienced employees leave. My goal is to try and predict if an employee will leave and to understand which features have the highest impact.

- For every employer, they want to ensure that they retain their employees for various reasons. Replacing a current employee would not only take time and resources, but the employer loses the institutional knowledge that has been placed in their employee for however long he/she worked for them. According to the Bureau of Labor Statistics, 5.2 million people separate from their jobs every month. Using different machine learning methods, I will try to answer the question: Why do employees leave?

DATA SET

Kaggle – Human Resources Analytics

- Features include:
  1) Satisfaction Level: Level of happiness at work. (1 = highest, 0 = lowest)
  2) Last Evaluation: Time that has past since the employee's last evaluation.
  3) Number of Projects: The number of projects an employee worked in during their tenure.
  4) Average Monthly Hours: On average, the number of hours an employee works a month.
  5) Time Spend with the Company: The number of years the employee was with their employer.
  6) Work Accident: Shows if the employee had a work related accident. (1 = yes, 0 = no)
  7) Left: Whether the employee left the company. (1 = yes, 0 = no)
  8) Promotion within the last 5 years: Whether or not an employee was promoted within the past five years.
  9) Department: The division the employee works in.
  10) Salary: Categorical level of how much an employee is paid. (low, medium, high)

PROCESS

EDA → KNN Model → Decision Tree → KNN (Cont.)

EXPLORATORY DATA ANALYSIS

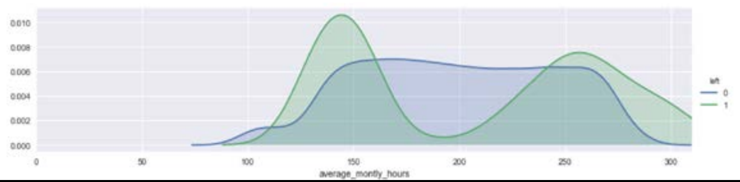
```
df.isnull().any()

satisfaction_level          False
last_evaluation             False
number_project              False
average_montly_hours        False
time_spend_company          False
Work_accident               False
left                        False
promotion_last_5years       False
sales                       False
salary                      False
dtype: bool
```

```
df.corr()
```

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company
satisfaction_level	1.000000	0.105021	-0.142970	-0.020048	-0.100866
last_evaluation	0.105021	1.000000	0.349333	0.339742	0.131591
number_project	-0.142970	0.349333	1.000000	0.417211	0.196786
average_montly_hours	-0.020048	0.339742	0.417211	1.000000	0.127755
time_spend_company	-0.100866	0.131591	0.196786	0.127755	1.000000
Work_accident	0.058697	-0.007104	-0.004741	-0.010143	0.002120
left	-0.388375	0.006567	0.023787	0.071287	0.144822
promotion_last_5years	0.025605	-0.008684	-0.006064	-0.003544	0.067433

```
#average_monthly_hours

g = sns.FacetGrid(df, hue="left",aspect=4)
g.map(sns.kdeplot,'average_montly_hours',shade= True)
g.set(xlim=(0, df['average_montly_hours'].max()))
g.add_legend()
```

<seaborn.axisgrid.FacetGrid at 0x117016d10>



Is Null

- The data set is clean
- No null values

Correlation

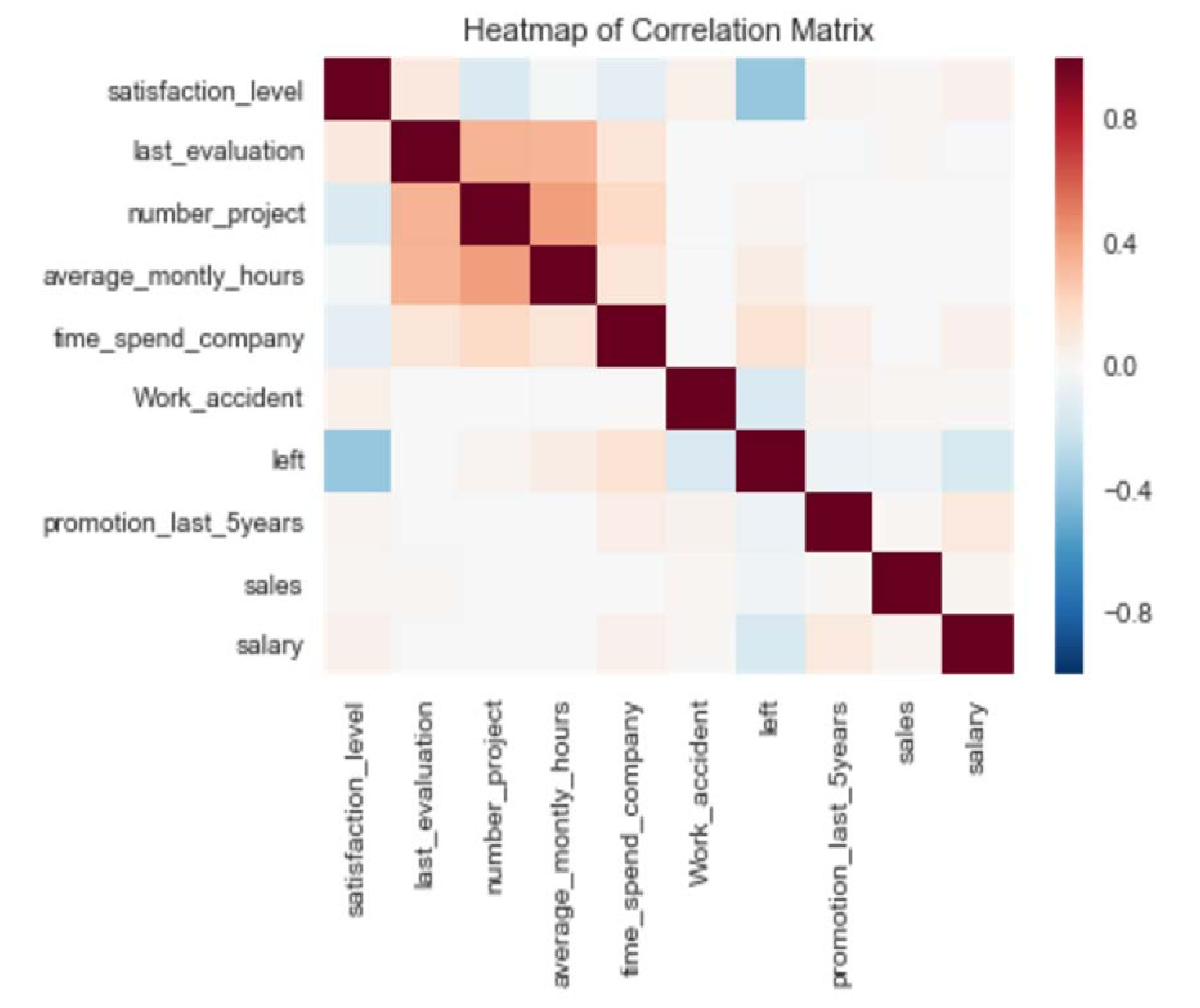
- A distribution of how correlated each data point is with each other
- Helps determine which features to select

Average Monthly Hours

- A visualization of average monthly hours compared to who left and stayed

HEATMAP

- Colorful heatmap
- Correlations between features used in the model
- Satisfaction level, number of projects, and average monthly hours worked have strong correlations with the employees who left



HYPOTHESIS AND METHOD

- **Hypothesis**
  - Based on HR data, a machine can predict whether or not an employee will leave better than the null accuracy. However, there will be outliers that fall into the false positives and false negatives of the model.

- **Method**
  - Producing a machine learning model that will take in features related to the employee and predict whether or not the current employees and new employees introduced to the model will leave or stay

MACHINE LEARNING MODELS

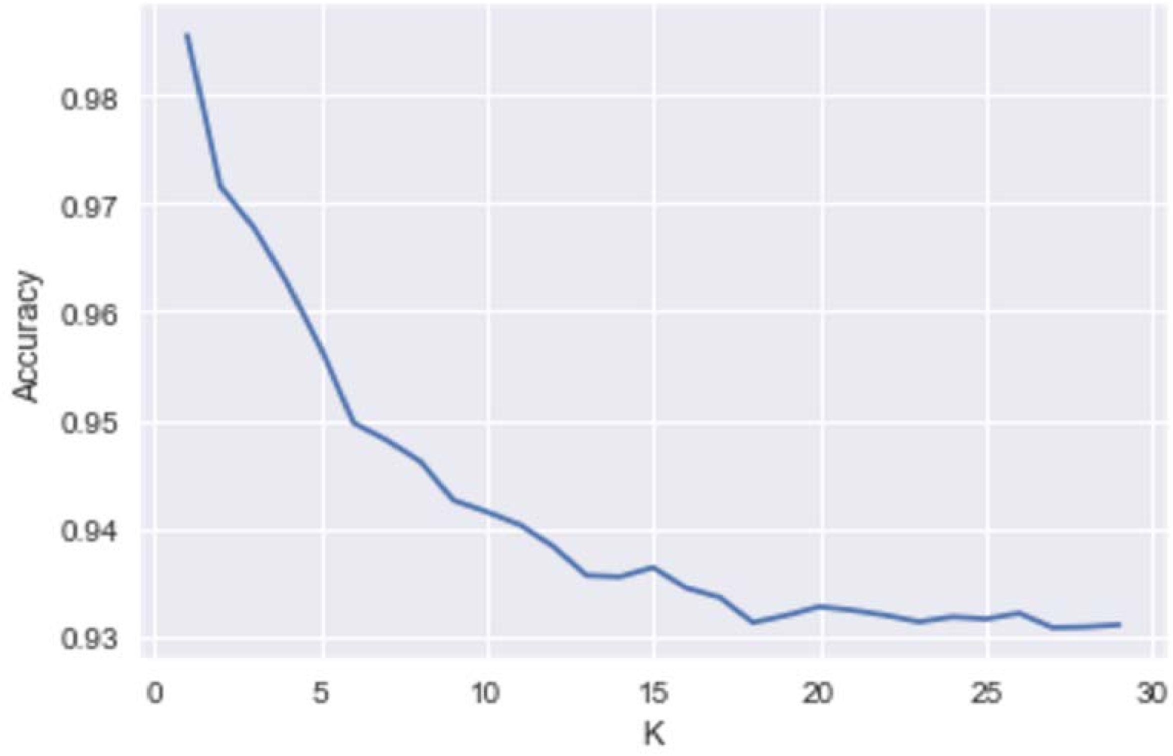
- **Initial Features Selected:** Satisfaction level, number of projects, and average monthly hours worked
- **Target (response):** Binary classification for employees who left (1) or stayed (0)
- **Null Accuracy (score to beat):** 0.68752187609
- **Train:** on current HR Analytics data set
- **Predict:** on current HR analytics data set

K-NEAREST NEIGHBOR

- Initial model was fitted with 1 neighbor, but it was not accurate
- Accuracy decreases as more neighbors are added.
- Focused on using the point at the elbow to enable a balance of overfit and accuracy to provide a broader generalization of the data

```
plt.plot(range(1,30), scores)
plt.ylabel("Accuracy")
plt.xlabel("K")
```

```
<matplotlib.text.Text at 0x11b319c50>
```



KNN – TEST | TRAIN | SPLIT

- Accuracy score when fitted from original dataset using 9 neighbors: 0.9425961730782052
- Accuracy score when training and testing with 80% of the data frame: 0.931733333333
- Cross Validation accuracy score: 0.94119768588122432
- Score is above the **null accuracy score**

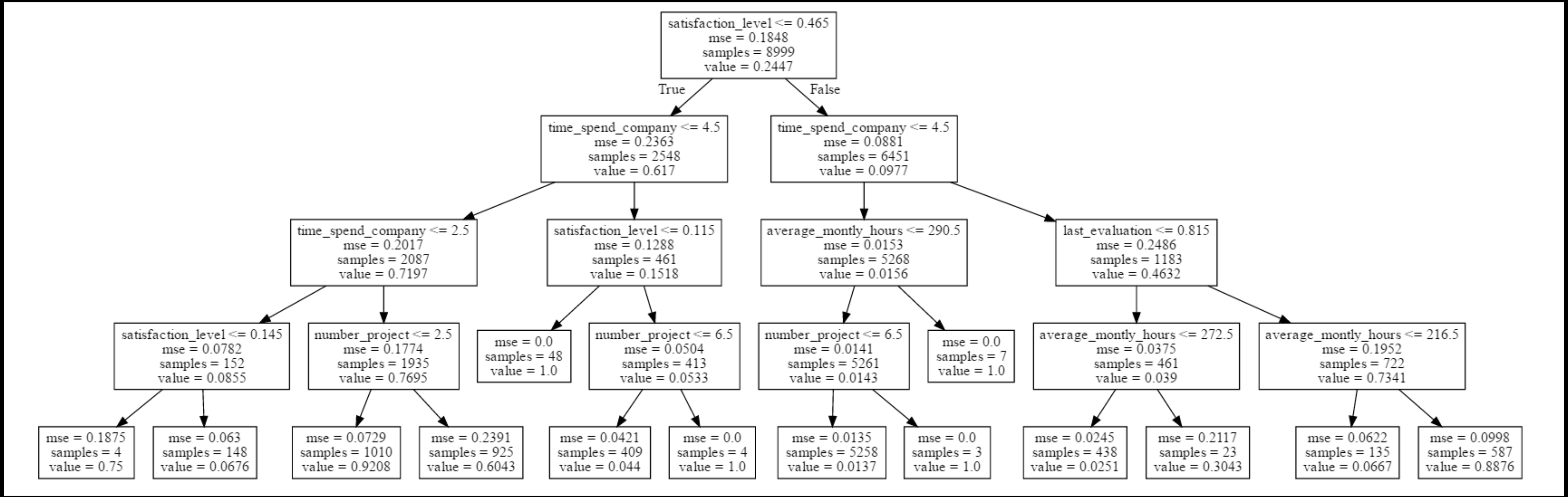
TEST | TRAIN | SPLIT (W/ SALARY)

- Accuracy score when fitted from original dataset using 9 neighbors: 0.9393959597306 4871
- Accuracy score when training and testing with 80% of the data frame: 0.9285333333333

- Cross Validation accuracy score: 0.94079730808118234

- Score is above the **null accuracy score,** but all of these scores are lower than initial model **without** salary

DECISION TREE



TREE - FEATURE SELECTION

```python
sorted(zip(model.feature_importances_, X.columns.values), reverse = True)
```

```
[(0.4487851453251805, 'satisfaction level'),
 (0.32109294420561602, 'time_spend_company'),
 (0.11445471049369017, 'last_evaluation'),
 (0.069441192343460939, 'average_montly_hours'),
 (0.046226007632054859, 'number_project'),
 (0.0, 'sales'),
 (0.0, 'salary'),
 (0.0, 'promotion_last_5years'),
 (0.0, 'Work_accident')]
```

- Initial model includes:
  - Satisfaction level, average monthly hours, and number of projects
- After creating a decision tree, see higher levels of correlation with different features

TEST | TRAIN | SPLIT (NEW MODEL)

- Accuracy score when fitted from original dataset using 9 neighbors: 0.9683312208147204
- Accuracy score when training and testing with 80% of the data frame: 0.9632
- Cross Validation accuracy score: 0.96686468713311347
- Score is above the **null accuracy score** and 2 other models with 2 new features

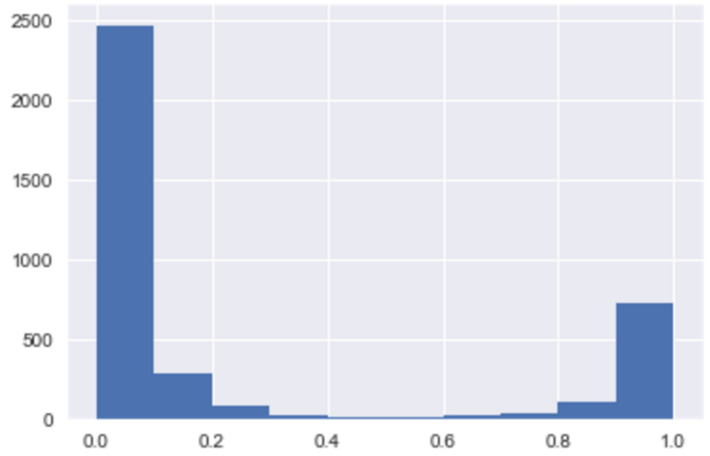
CONFUSION MATRIX AND TUNING

```
#Updated Confusion Matrix - Visual of Cross Validation Score

from nltk import ConfusionMatrix
preds = np.where(probs > .1, 1, 0)
print ConfusionMatrix(list(y_test), list(preds))
```

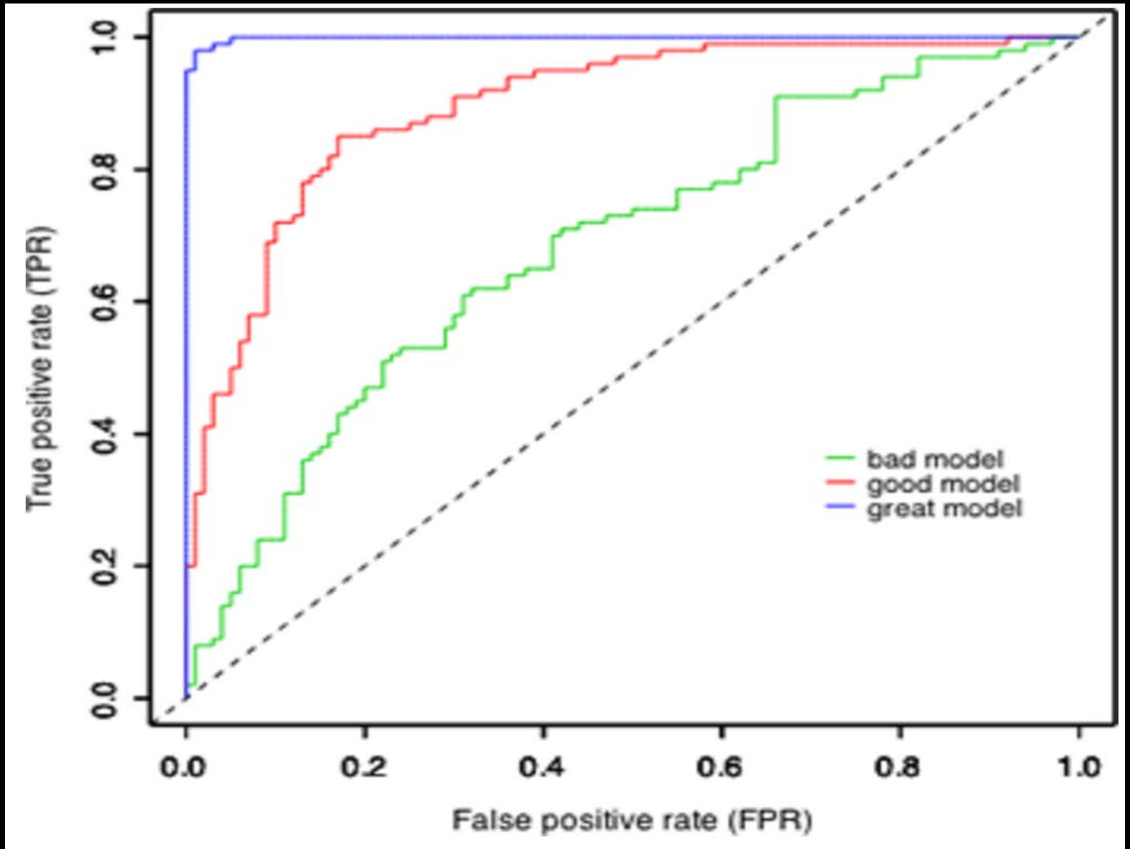
```
  |    0    1 |
--+-----------+
0 |<2450> 403 |
1 |   16 <881>|
--+-----------+
(row = reference; col = test)
```

```
#Probability of predictions from X_test data

probs = knn.predict_proba(X_test)[:, 1]
plt.hist(probs)
```

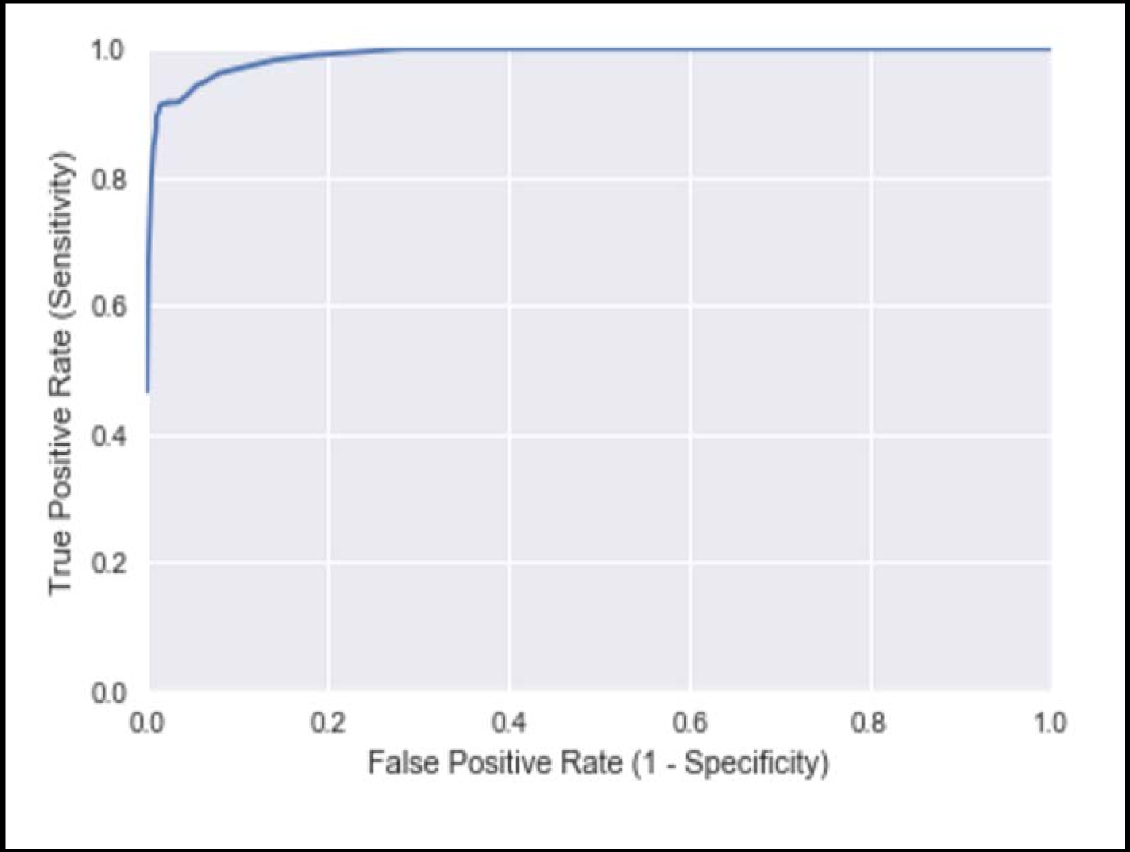
```
(array([ 2466.,   279.,    79.,    27.,    11.,     4.,    19.,    31.,
          108.,   726.]),
 array([ 0. ,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6,  0.7,  0.8,  0.9,  1. ]),
 <a list of 10 Patch objects>)
```



ROC CURVE – MODEL EVALUATION



Class



Project

CONCLUSION

- Satisfaction level is the most correlated feature in the data set
- 3 features made up ~ 88% of the most “important” features to split on for the decision tree (satisfaction level, time spent with company, last evaluation)
- Data was not complete, salary was an example where creating categories based on arbitrary categories can skew some of the prediction results

NEXT STEPS

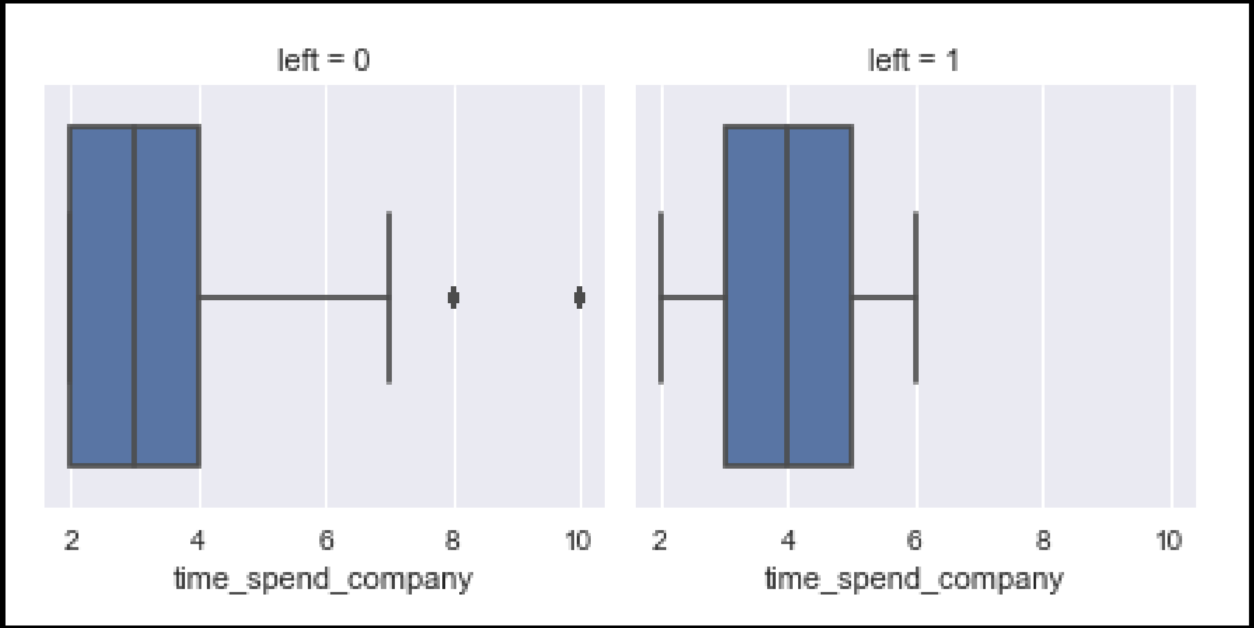
- Understanding and determining different combinations of features and their relationship with each other
- Work on tuning the confusion matrix to train the model to get closer results I am specifically looking for
- Use a gini index to evaluate and optimize the decision tree

APPENDIX

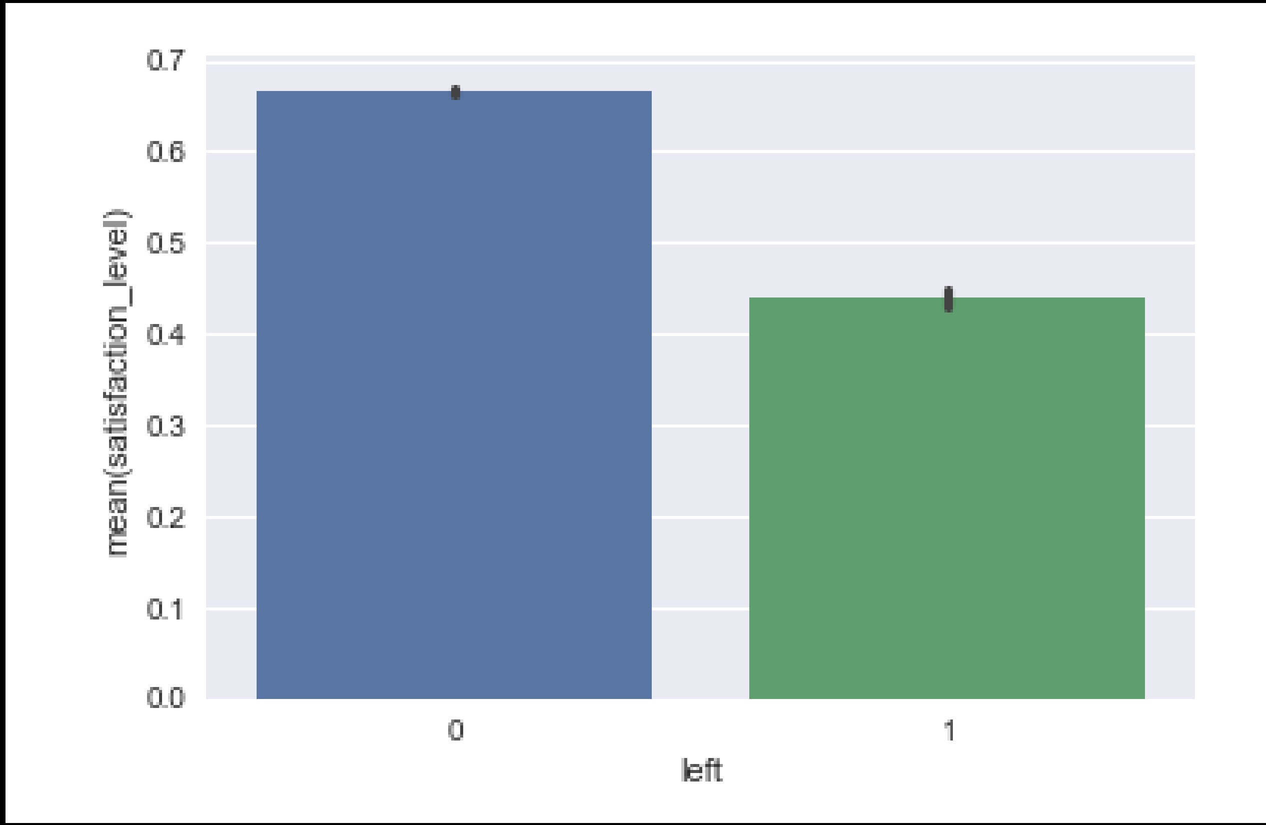
REFERENCES

- Kaggle Data Set
- Bureau of Labor Statistics

TIME SPENT WITH EMPLOYER



SATISFACTION LEVEL



AVERAGE MONTHLY HOURS