MONTCLAIR STATE UNIVERSITY

Feliciano School of Business

EMPLOYEE TURNOVER

CASE STUDY

INFO 570 Data Wrangling and Analysis

TEAM 11

Anace Alhashhoush

Omar Altaher

Amanda Maguire

Ali Abughazaleh

INTRODUCTION

- This project is done using information from TECHCO Company to explore the significant causes behind its employee turnover

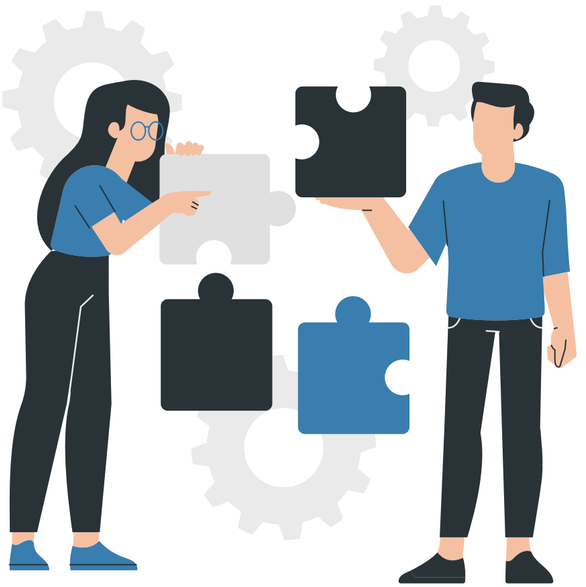
TECHCO

- TECHCO is a real technology company in India

- TECHCO model's strengths: a combination of value model, technological model, distribution model, and financial model

BUSINESS PROBLEM

- Employee turnover is a weakness and a problem for this firm demonstrating that a lack of spotting trends in TECHCO is a key barrier that must be overcome
- The HR department being unable to improve the capabilities of employees to match the requirements of the fast pace and contemporary technology due to employee turnover and not attacking its causes



EMPLOYEE TURNOVER

- Employee turnover refers to the total number of workers who leave a company over a certain time period
- It includes those who exit voluntarily as well as employees who are fired or laid off—that is, involuntary turnover

EMPLOYEE TURNOVER IMPORTANCE

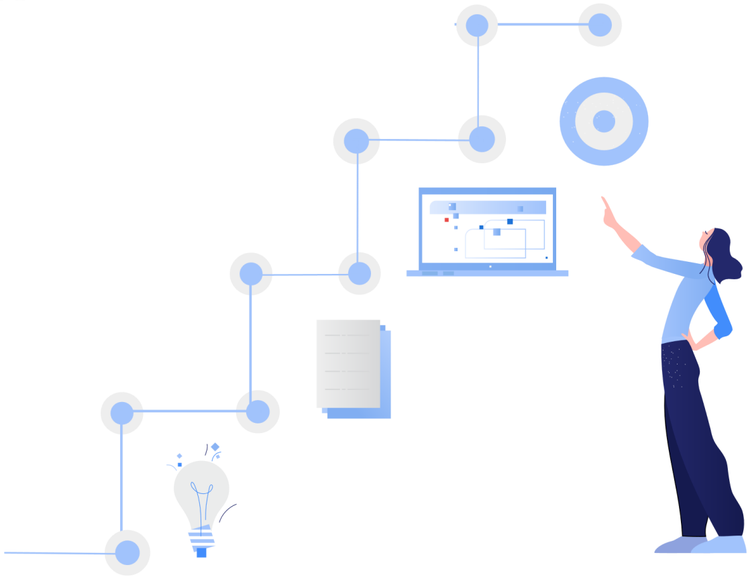
- Employee turnover is extremely important to an organization
- When an employee leaves the company, it costs the firm both time and money
- Costs Include:
  - Hiring a replacement
  - Training for the position
  - Decreased productivity
  - Increased monetary costs associated with lower productivity
- High turnover rates can reflect negatively on businesses, making it difficult to attract and retain top talent
- Decreasing turnover, is a business strategy each company should follow

PROJECT PRIMARY GOAL

Exploring patterns and spotting trends in the data that will help us learn more about the drivers of employee turnover to help the HR department understand the causes behind the company employee turnover, thus they can better manage their strategies to make better decisions in retaining employees

EXPECTED OUTCOMES

- Reduce or eliminate high turnover rates resulting in increased productivity
- Reduce monetary costs associated with lower productivity
- Reduce time spent training and hiring employees



PROJECT PLAN

Using Python

Analyzing and refining data of 1,191 entry-level employees

Describing Data

Describing the dataset which will help us learn more about the drivers of employee turnover

Data preprocessing

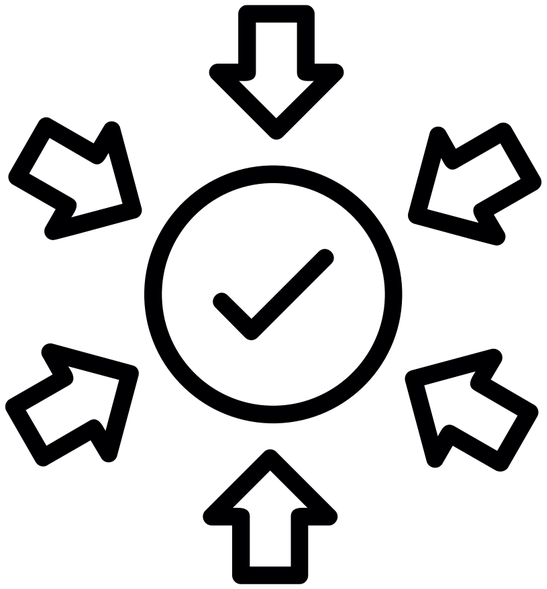
Variable treatment, dropping columns and rows when necessary, inputting missing values, and transforming variables if necessary

Best Model

Exploring patterns and determining the best predictive model to predict turnover

PROJECT IMPLICATIONS

- Discover nonlinear and interactive patterns between variables that may otherwise have gone unnoticed
- Predict when current employees may leave so they can be proactive in hiring new employees
- Explore patterns and spot trends in the data that will help us learn more about the drivers of employee turnover
- These elements coming together can serve as the basis to build a solid tech business model



TECHCO DATASET

- The dataset was collected from Employee Turnover at TECHCO chosen from Kaggle and distributed using a excel spreadsheet

- The data are structured as a panel with one observation for each month that an individual is employed at the company for up to 40 months

- The data include 34,453 observations from 1,191 employee's total

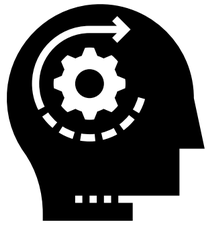
- The employees were quasi-randomly deployed to any of TECHCO's nine geographically dispersed production centers in 2007

Using Kaggle link:
https://www.kaggle.com/datasets/ryanthomasallen/simulated-data-for-ml-paper

DATA INPUT

Time
Number of months at the company

Training Score
Employees' performance scores in an intensive three-month on-boarding training course

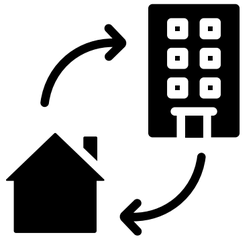


Logical Score
Standardized university exit exam score - logical section

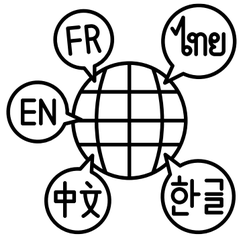
Verbal Score
Standardized university exit exam score - verbal section

Average Literacy
Average literacy in the employee's home region

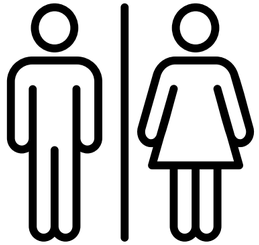
Location Age
The age of the employee's assigned production center



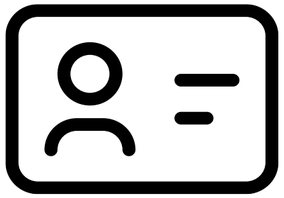
Distance
Distance between the employee's home and their quasi-randomly assigned production center



Similar Language
The similarity of the prevailing language in the production center's region to that of the employee's hometown

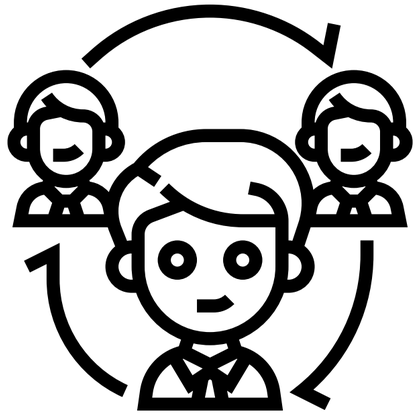


Gender
Male = 1, Female = 0



Employee Id
Identifying number for each employee

DATA OUTPUT



Turnover

Indicates whether the employee left or stayed during that time period(40 months)

DESCRIPTIVE ANALYSIS

- Rows: 34452
- Columns: 11
- Zero percent of missing values

```
df.head()
```

	time	training_score	logical_score	verbal_score	avg_literacy	location_age	distance	similar_language	is_male	emp_id	turnover
0	1	4.840446	5	2	81.05207	6	1.635494	24.11053	1	1	Stayed
1	2	4.840446	5	2	81.05207	6	1.635494	24.11053	1	1	Stayed
2	3	4.840446	5	2	81.05207	6	1.635494	24.11053	1	1	Stayed
3	4	4.840446	5	2	81.05207	6	1.635494	24.11053	1	1	Stayed
4	5	4.840446	5	2	81.05207	6	1.635494	24.11053	1	1	Stayed

• NUMERICAL & CATEGORICAL VARIABLES

```
numerical_var = list(df.select_dtypes(exclude=object).columns)
# Filtering columns that are only numerical
numerical_var
```

```
['time',
 'training_score',
 'logical_score',
 'verbal_score',
 'avg_literacy',
 'location_age',
 'distance',
 'similar_language',
 'is_male',
 'emp_id']
```

```
categorical_var =  list(df.select_dtypes(object).columns) #selecting categorical varibles only
categorical_var
```

```
['turnover']
```

DATA TYPES

```
df.dtypes
```

```
time                 int64
training_score     float64
logical_score        int64
verbal_score         int64
avg_literacy       float64
location_age         int64
distance           float64
similar_language   float64
is_male              int64
emp_id               int64
turnover            object
dtype: object
```

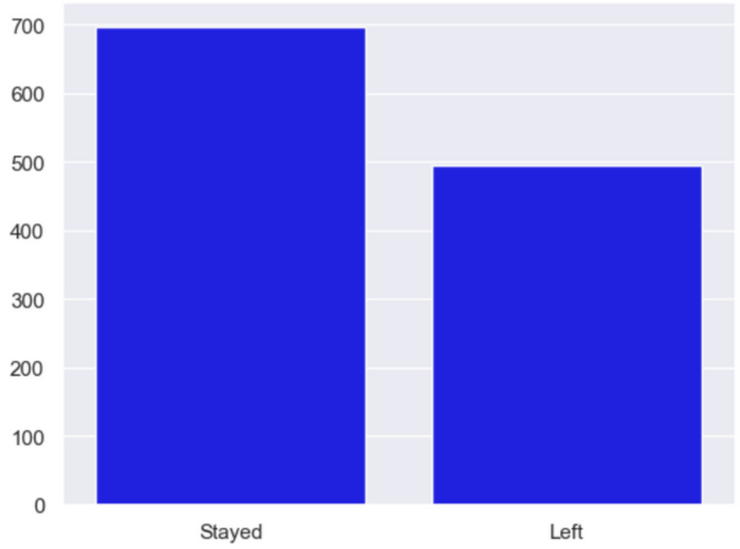
• PIVOT TABLES | TURNOVER (STAYED OR LEFT)

```
# count of employees who stayed or left
pd.pivot_table(df_last, values='time', index=['turnover'], aggfunc = 'count')
```

turnover	time
Left	494
Stayed	697

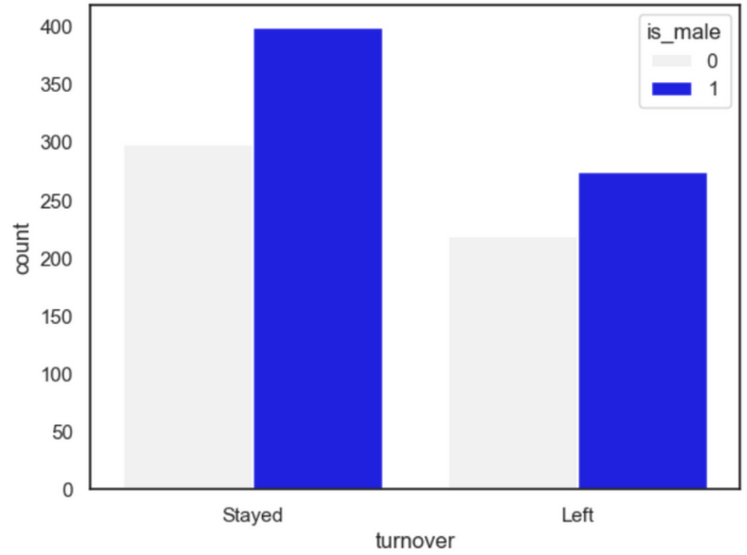
• BARPLOT (STAYED | LEFT)

```
sns.set(style="darkgrid")
ax = sns.barplot(x=df_last['turnover'].value_counts().index,
                 y=df_last['turnover'].value_counts().values,
                 color = 'blue')
```



• COUNTPLOT (TURNOVER | IS-MALE)

```
# stayed vs left employees, hue for gender
sns.set(style="white")
ax = sns.countplot(x='turnover', hue = 'is_male', data = df_last,
                   color = 'Blue')
```



• COUNT| MEAN | STD | MIN | MAX | VARIANCE

	emp_id	time	training_score	logical_score	verbal_score	avg_literacy	location_age	distance	similar_language	is_male
count	1191.000000	1191.000000	1191.000000	1191.000000	1191.000000	1191.000000	1191.000000	1191.000000	1191.000000	1191.000000
mean	596.000000	28.926952	4.392502	4.473552	4.649874	75.712149	15.222502	0.816037	59.643230	0.565911
std	343.956393	10.982440	0.544875	3.928876	4.438657	9.273565	7.939056	0.750328	35.303239	0.495845
min	1.000000	1.000000	2.688673	-5.000000	-7.000000	49.354540	2.000000	0.000000	1.250000	0.000000
25%	298.500000	24.000000	4.171338	1.000000	1.000000	68.657835	9.000000	0.199370	27.170575	0.000000
50%	596.000000	34.000000	4.524776	4.000000	4.000000	77.161420	11.000000	0.579961	55.704960	1.000000
75%	893.500000	37.000000	4.809570	8.000000	7.000000	82.838030	24.000000	1.251390	98.798635	1.000000
max	1191.000000	39.000000	5.110679	12.000000	17.000000	97.357410	28.000000	3.200019	100.000000	1.000000

Note: in our dataset we found no ZERO VARIANCES since no standard deviation is equal to ZERO and the variance is the square of the standard deviation

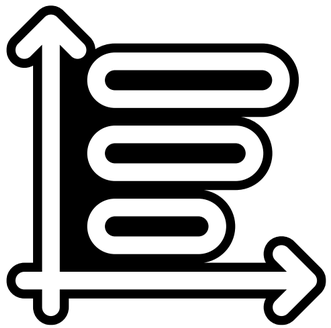
• MODE | MEDIAN | MISSING VALUES

```
df_last.mode()[:1]
```

	emp_id	time	training_score	logical_score	verbal_score	avg_literacy	location_age	distance	similar_language	is_male	turnover
0	1	37.0	4.682977	0.0	0.0	83.58855	10.0	0.0	100.0	1.0	Stayed

```
df_last.median()
```

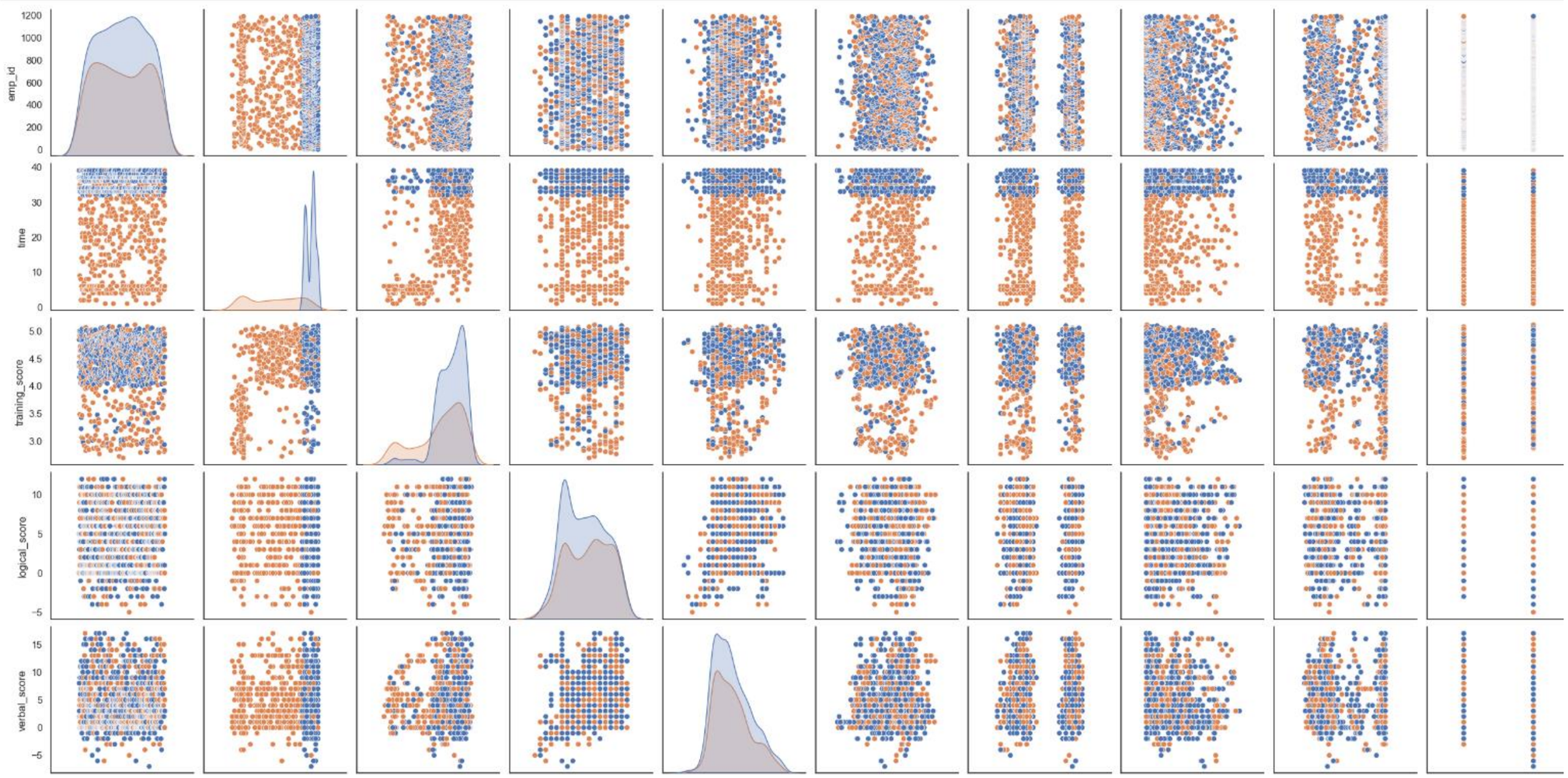
```
emp_id              596.000000
time                 34.000000
training_score        4.524776
logical_score         4.000000
verbal_score          4.000000
avg_literacy         77.161420
location_age         11.000000
distance              0.579961
similar_language     55.704960
is_male               1.000000
dtype: float64
```



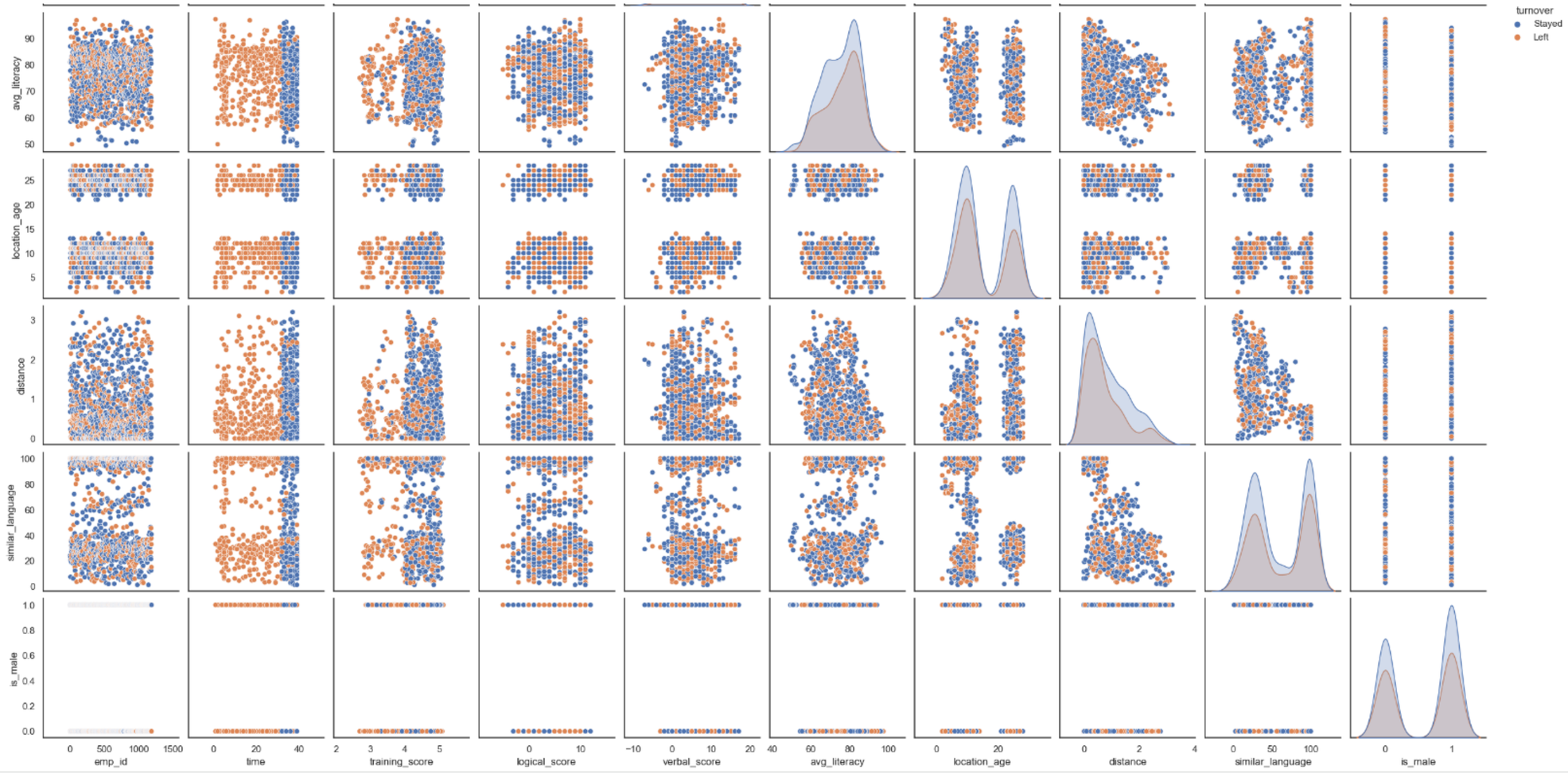
```
# create a DataFrame indicating which values are missing
missing_value_df = df.isnull()

# calculate the percentage of missing values for each column
percent_missing = missing_value_df.mean()
percent_missing
```

```
time                0.0
training_score      0.0
logical_score       0.0
verbal_score        0.0
avg_literacy        0.0
location_age        0.0
distance            0.0
similar_language    0.0
is_male             0.0
emp_id              0.0
turnover            0.0
dtype: float64
```

• SCATTERPLOTS



• SCATTERPLOTS CONT.



DATA PREPROCESSING

- Preprocessing the data to ensure accuracy and high quality data
- Before creating a correlation matrix or the heat map we need to clarify that the data was not linear and we scaled and then standardized the data to become linear
- Due to zero percent missing values, no values needed to be inputted
- Using Scaling, Standardization, and SMOTE technique to transform the data into becoming linear
- Duplicate data were removed (34452 -> 1191)

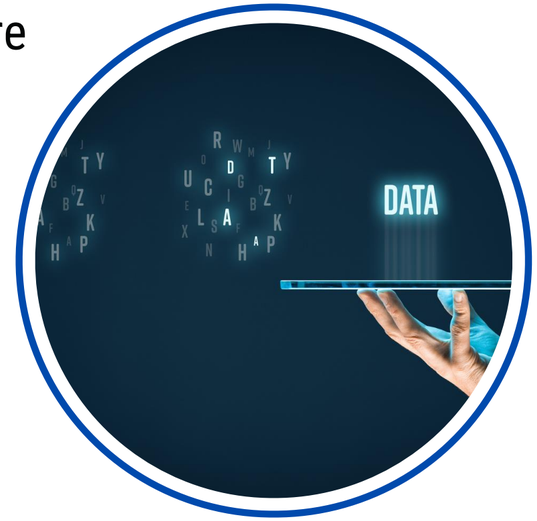
• DATA SCALING

- One reason we may want to scale a feature before standardizing is that some machine learning algorithms are sensitive to the scale of the input features

- By scaling the features to a common scale, we can improve the performance of these algorithms. Standardizing the scaled features can further fine-tune the results and ensure that the features have a mean of 0 and a standard deviation of 1

- The purpose of scaling is to normalize the data within a range

- Below we extracted numerical values scaling and overwriting the data



	emp_id	time	training_score	logical_score	verbal_score	avg_literacy	location_age	distance	similar_language	is_male	turnover
0	1	39	4.840446	5	2	81.05207	6	1.635494	24.11053	1	Stayed
1	2	34	4.355449	0	8	93.72386	3	0.234684	96.08640	1	Stayed
2	3	37	4.416302	3	2	66.49519	6	0.673065	100.00000	1	Stayed
3	4	34	4.995522	11	3	87.05980	13	0.232718	93.20673	1	Stayed
4	5	33	4.531571	1	0	77.78675	24	2.333878	23.03190	0	Stayed

DATA STANDARDIZING

- Preprocessed the data using standardization
- The purpose of standardization is to scale the values so the mean is 0 and standard deviation is 1

The steps we used to standardize:

- Created column names for standardized values
- Converted numerical values to numpy array
  -Numpy array fast and efficient python sequence
- Created standardization instance
  -Gathering data together into bundles
- Standardize the numerical values

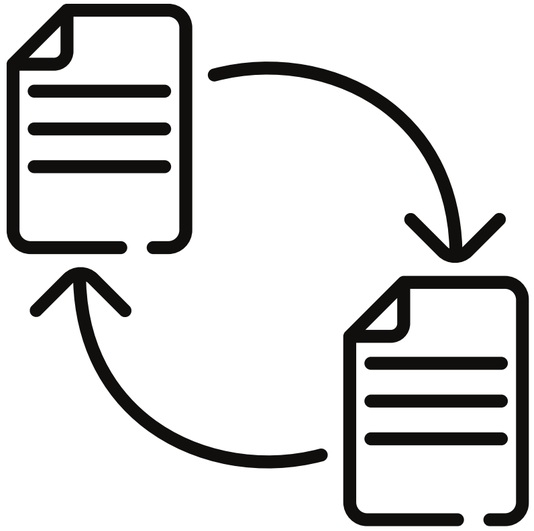
UTILIZING DUMMY VARIABLES

- Preprocessed the data converting categorical variables to dummy variables

- The purpose of dummy variables is to display the categorical values as numerical

  -Important step to avoid errors in the functions

  -Ultimately dropping any categorical variable strings



OUTLIERS

- Preprocessed the data using the z-score function
- The purpose is to find outliers based on the z-score after scaling and standardizing the data
- Our threshold value used was z >3
- 99.7% of the data points lie between +/- 3 standard deviation

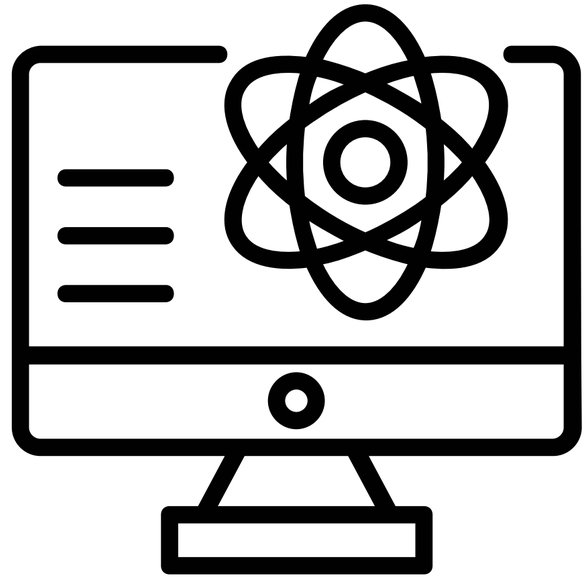
Between all variables values, no outliers were found!

	similar_language_standardized	time_standardized	\
0	False	False	
1	False	False	
2	False	False	
3	False	False	
4	False	False	
...	...	...	
1389	False	False	
1390	False	False	
1391	False	False	
1392	False	False	
1393	False	False	

	training_score_standardized	location_age_standardized	\
0	False	False	
1	False	False	
2	False	False	
3	False	False	
4	False	False	
...	...	...	
1389	False	False	
1390	False	False	
1391	False	False	
1392	False	False	
1393	False	False	

• USING THE SMOTE TECHNIQUE

- Preprocessed the data using the Smote Technique
- The purpose is to use oversampling to balance the data for random samples
- Input variable being Turnover_Stayed



BEFORE

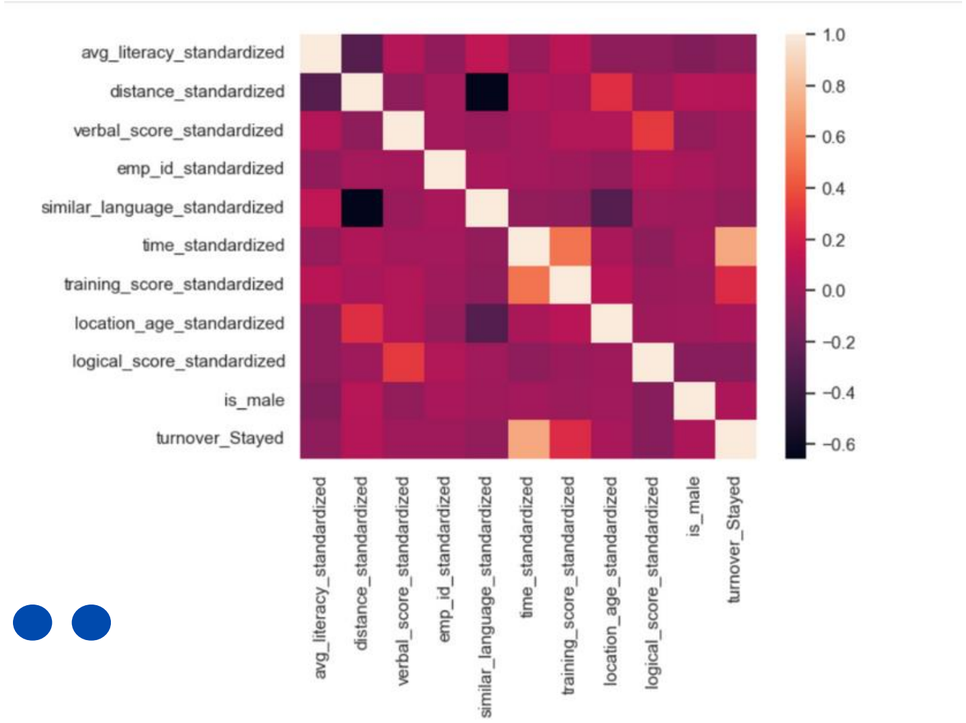
```
In [27]: df_last['turnover_Stayed'].value_counts()
Out[27]: 1    697
         0    494
         Name: turnover_Stayed, dtype: int64
```

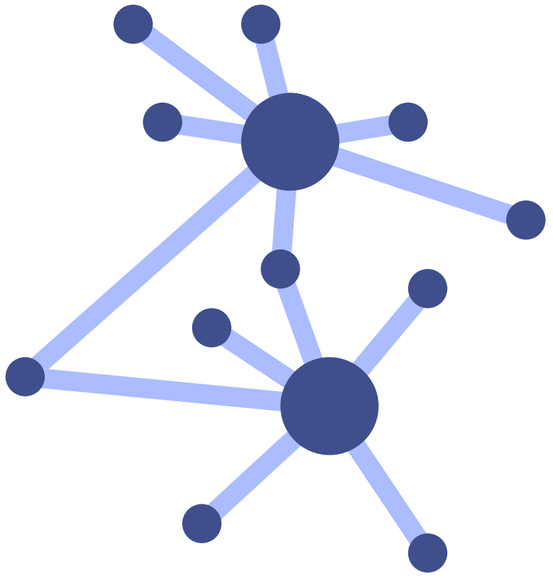
AFTER

```
In [29]: df_last['turnover_Stayed'].value_counts()
Out[29]: 1    697
         0    697
         Name: turnover_Stayed, dtype: int64
```

• PREPROCESSING COMPLETE; TIME TO CORRELATE!

- The data is now at its highest, most valuable form
- Using this data, we calculated pairwise correlation between all columns
- Heatmap to display the results



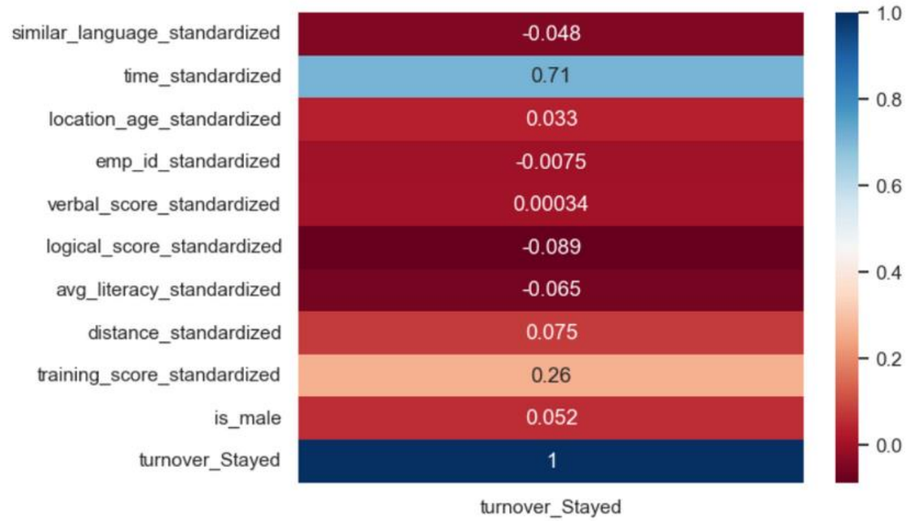


• CORRELATION

	similar_language_standardized	time_standardized	location_age_standardized	emp_id_standardized	verbal_score_standardized	logical_score_standardized	avg_literacy_standardized	distance_standardized	training_score_standardized	is_male	turnover_Stayed
similar_language_standardized	1.000000	-0.044886	-0.292289	0.033590	-0.018296	0.007667	0.129651	-0.656223	-0.060774	-0.004065	-0.048034
time_standardized	-0.044886	1.000000	0.044012	0.022952	0.015175	-0.064862	-0.028508	0.062614	0.518934	0.013290	0.710819
location_age_standardized	-0.292289	0.044012	1.000000	-0.035345	0.073595	0.002006	-0.063287	0.270503	0.097534	0.004396	0.033396
emp_id_standardized	0.033590	0.022952	-0.035345	1.000000	0.011420	0.070345	-0.050947	0.023225	-0.000437	0.032696	-0.007506
verbal_score_standardized	-0.018296	0.015175	0.073595	0.011420	1.000000	0.315893	0.080017	-0.063381	0.072149	-0.047180	0.000336
logical_score_standardized	0.007667	-0.064862	0.002006	0.070345	0.315893	1.000000	-0.065278	-0.007199	-0.016652	-0.088864	-0.088640
avg_literacy_standardized	0.129651	-0.028508	-0.063287	-0.050947	0.080017	-0.065278	1.000000	-0.285284	0.103388	-0.119205	-0.065218
distance_standardized	-0.656223	0.062614	0.270503	0.023225	-0.063381	-0.007199	-0.285284	1.000000	0.033434	0.092527	0.075208
training_score_standardized	-0.060774	0.518934	0.097534	-0.000437	0.072149	-0.016652	0.103388	0.033434	1.000000	-0.011331	0.261164
is_male	-0.004065	0.013290	0.004396	0.032696	-0.047180	-0.088864	-0.119205	0.092527	-0.011331	1.000000	0.051876
turnover_Stayed	-0.048034	0.710819	0.033396	-0.007506	0.000336	-0.088640	-0.065218	0.075208	0.261164	0.051876	1.000000

Correlation and Turnover_Stayed

- After processing the correlation between all columns we looked at the correlation between the columns and Turnover_Stayed



BEST MODEL

Examining Variables before Fitting Models

- Examined emp_id_standardize to ensure there were no repeated values
- Duplicate values have been dropped, ensuring the latest data for each employee
- As a reminder, emp_id_standardize is a number given to each employee
- Resulted in no repeated values, confirming the count of each employee is one time

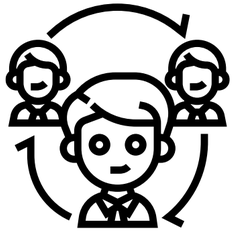
Removed low correlated variables

- Based on the displayed variable values, we kept those variables for the best model and removed the rest

Variable	Value
Time_standardized	0.71
Training_score_standardized	0.26
Turnover_Stayed	1.0

Training Score
Employees' performance scores in an intensive three-month on-boarding training course

Time
Number of months at the company



Turnover

• FITTING MODELS USING R^2

- In order to find the best model, we built a dataframe containing the target values printing the R^2

-The R^2 value displayed the percent of variance

-We aimed for the highest percentage possible

-The higher the R^2, the better the model

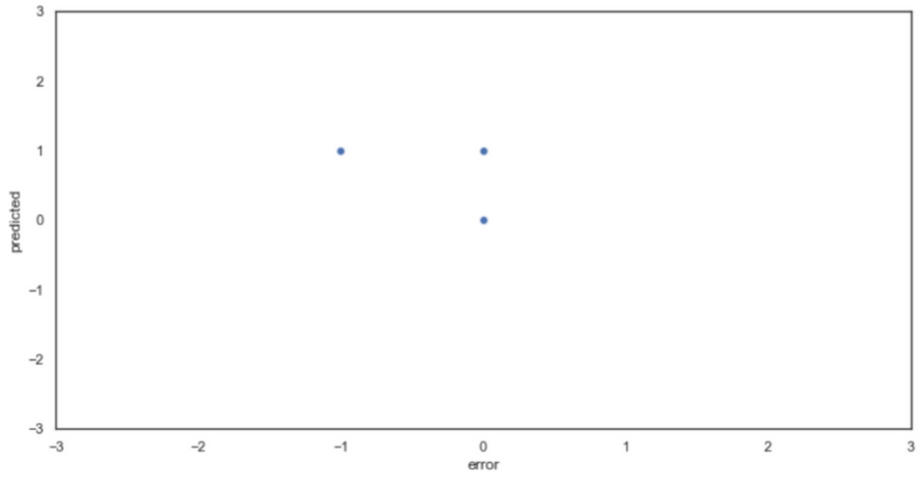
- We landed at a R^2 value of 52.11
- Model coefficients: [0.3863672 , -0.07194487]

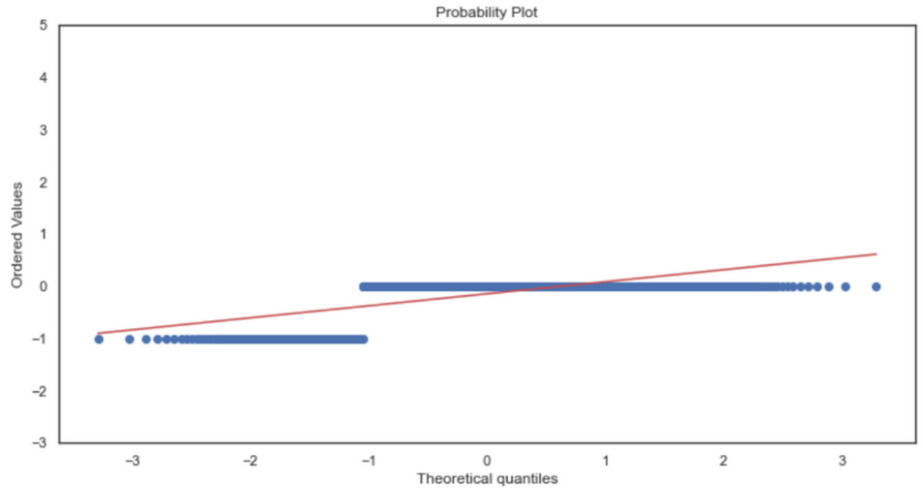
RESULTS PRINTED

	actual	predicted	error
0	1	1.0	0.0
1	1	1.0	0.0
2	1	1.0	0.0
3	1	1.0	0.0
4	1	1.0	0.0

ASSESSING THE MODEL BY PLOTTING

- Used a scatterplot to display the error vs. the predicted values
- Set the parameters of the x and y axis at (-3, 3)
- Used a normality plot to display the theoretical quantities vs. the ordered values
- Set the parameters of the x and y axis at (-3, 5)





• CHOOSING A FINAL MODEL

Evaluated the four models built

- Logistic Regression
- Decision Tree
- Random Forest
- K-NN

Measurement performance criteria - The higher the value the better result!

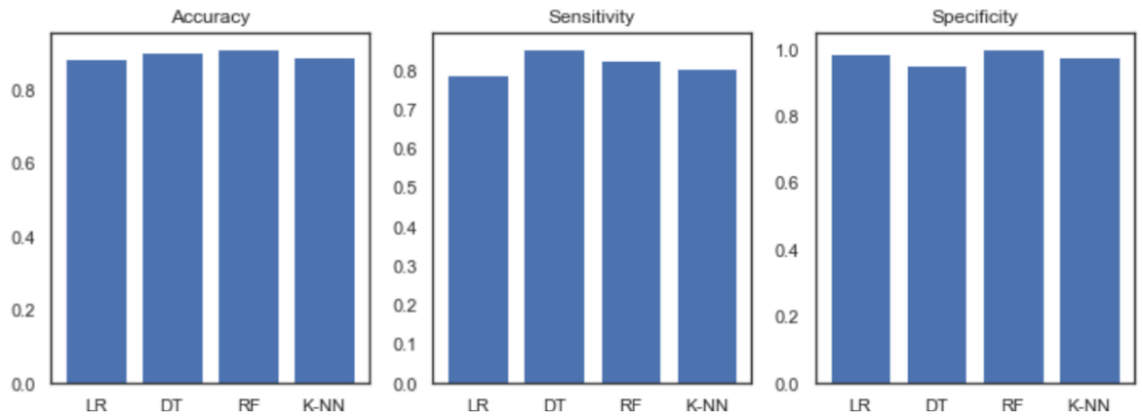
- Accuracy - how often a model predicts the right class
- Sensitivity - how the model detects positive cases
- Specificity - how the model detects negative cases

```
# computes the confusion matrix and performance indicators
def get_performance(actual_Y, pred_Y):
    cm = confusion_matrix(actual_Y, pred_Y)
    total = sum(sum(cm))
    accuracy = (cm[0,0]+cm[1,1])/total
    sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
    specificity = cm[1,1]/(cm[1,0]+cm[1,1])
    return accuracy, sensitivity, specificity
```

• RESULTS OF THE MODELS BUILT

The four models built and their measurement values

Models/Measurements	Accuracy	Sensitivity	Specificity
Logistic Regression	0.887202	0.789700	0.986842
Decision Tree	0.902386	0.854077	0.951754
Random Forest	0.911063	0.824034	1.000000
K-NN	0.891540	0.806867	0.978070



- Random Forest produced the highest measurement values in two of the performance indicators
- By a margin, we concluded that Random Forest is the best model when looking at employee turnover

Model Code:

- RF = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=0)
- RF.fit(X_train, Y_train)

KEY MODEL CONCLUSIONS

- In fitting models, we concluded that our target variable is employee turnover, specifically turnover_stayed

- Based on the correlation matrix, we concluded that training_score and time are the most important variables contributing to turnover at TECHCO

- Due to the model measurement values, we concluded that Random Forest is the best model to predict turnover

- This means we can look at the employee's training_score and time values at TECHCO using the Random Forest model to predict turnover!

• CODES TO PREDICT TURNOVER

Code to enter the variable values for time and training_score

```
X_new=[]
mean = [mean_t, mean_tr]
std = [std_t, std_tr]
new = []
n = 2
for i in range(0, n):
    if i == 0:
        print("Enter the time (in months):")
    else:
            print("Enter the training_score: ")
    ele = input()
    new.append(ele)
for i in range(0, len(new)):
    new[i] = float(new[i])
new[0] = (new[0] - min(df['time']))/max(df['time'])
new[1] = (new[1] - min(df['training_score']))/max(df['training_score'])
for i1, i2, i3 in zip(mean, std, new):
    X_new.append((i3-i1)/i2)
X_new
```

Code to return the result or turnover_stayed based on the values

```
# Create a scaling and standardization object
scaler = StandardScaler()

# Fit the scaling and standardization object to the original data
scaler.fit(X_train)

# Apply the scaling and standardization transformations to the original data
X_train_scaled = scaler.transform(X_train)

# Create a Random Forest model and fit it to the scaled data
model = RandomForestClassifier()
model.fit(X_train_scaled, Y_train)

X_new = np.array(X_new).reshape(-1, 2)

# Use the model to make predictions on the scaled new data
predictions = model.predict(X_new)

predictions
```

SCENARIOS

- The HR department at TECHCO is preparing for their upcoming interviews with prospective employees

- In order to know how many employees will need to be replaced, they look at the current employee characteristic values to predict turnover

- Will they stay or leave TECHCO and need to be replaced?

Employee_Id	Time (Months)	Training_Score	Result
Employee 1	4	2	LEAVE
Employee 2	15	3	LEAVE
Employee 3	35	4	STAY

CONCLUSION

- The HR department at TECHCO is now fully equipped to determine if an employee will stay or leave the organization

- This has many benefits for anticipating turnover at the company and being proactive about it

- This is a real business problem in the Tech Industry

  -Forbes.com states that the average turnover in Tech is 13.2%
  -Out of every business sector, Tech has the HIGHEST turnover rate

- Using machine learning evaluating patterns and trends, we created a model to inform TECHCO if an individual employee will leave, resulting in turnover

TEAM 11

THANK YOU