



CPSC 481: ARTIFICIAL INTELLIGENCE

EMPLOYEE HAPPINESS MONITORING PROJECT

- BY SHIVANGI SHAKYA



Topics To Discuss



MOTIVATION



APPROACH



PROGRAMMING LANGUAGE



DATASETS



CODE FLOW



DEMONSTRATION



POSSIBLE EXTENSIONS



RESULTS



RESOURCES

MOTIVATION

MENTAL HEALTH :-

- Individual and organizational growth.
- Aware employers about the employee mental health status.
- Employee happiness score derivation based on factors' computation.
- End goal:
 - create the system for monitoring the employee's health status
 - maintain the positive and stress-free work environment in companies.





APPROACH



Factors: Train Input Data

- *Working Duration*
- *Gender*
- *Type of Company*
- *Work From Home*
- *Designation*
- *Resource Allocation*
- *Mental Fatigue*



Trained Model



OUTPUTS:

- **Happiness Predicted Value**
- **Happy Or Unhappy?**



Test Input Data



PROGRAMMING LANGUAGE

- Python
 - Numpy
 - Pandas
 - Scikit Learn Library
 - Gaussian Naïve Bayes Classifier

DATASETS

Train Data:

- [train.csv](#)

Out[124]:

	Employee ID	Date of Joining	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score	Happiness
0	ffe32003000360033003200	9/30/08	Female	Service	No	2	3.0	3.8	0.84
1	ffe3700360033003500	11/30/08	Male	Service	Yes	1	2.0	5.0	0.64
2	ffe31003300320037003900	3/10/08	Female	Product	Yes	2	NaN	5.8	0.51
3	ffe32003400380032003900	11/3/08	Male	Service	Yes	1	1.0	2.6	0.80
4	ffe31003900340031003600	7/24/08	Female	Service	No	3	7.0	6.9	0.48

Test Data:

- [test.csv](#)

Out[168]:

	Employee ID	Date of Joining	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score
0	ffe31003300390039003000	12/10/08	Female	Service	No	2	5	7.7
1	ffe31003300310037003800	8/14/08	Female	Product	Yes	1	2	5.2
2	ffe33003400380035003900	11/13/08	Male	Product	Yes	1	3	5.9
3	ffe3100370039003200	2/7/08	Female	Service	No	3	6	4.6
4	ffe32003600390036003700	7/17/08	Female	Product	No	2	5	6.4

CODE FLOW:

- Creating new column [HappyCond] using below condition :

```
In [125]: data_train['HappyCond'] = np.where(data_train["Happiness"]>=0.5,1,0)
data_train.head()
```

Out[125]:

	Employee ID	Date of Joining	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score	Happiness	HappyCond
0	fffe32003000360033003200	9/30/08	Female	Service	No	2	3.0	3.8	0.84	1
1	fffe3700360033003500	11/30/08	Male	Service	Yes	1	2.0	5.0	0.64	1
2	fffe31003300320037003900	3/10/08	Female	Product	Yes	2	NaN	5.8	0.51	1
3	fffe32003400380032003900	11/3/08	Male	Service	Yes	1	1.0	2.6	0.80	1
4	fffe31003900340031003600	7/24/08	Female	Service	No	3	7.0	6.9	0.48	0

- Creating new column [tenure] from 'Date of Joining' .

```
data_train['tenure'] = data_train['today'] - data_train["Date of Joining"]
data_test['tenure'] = data_test['today'] - data_test["Date of Joining"]
```

- Drop those rows where [happiness] is empty.

```
# Drop rows where target variable "Happiness" is missing.
data_train = data_train.dropna(subset=['Happiness'])
```

- Impute remaining missing values with medians.
- Train a Gaussian Naive Bayes classifier on the training set.
- Test the model using test.csv.

RESULT

- Model accuracy score: 0.9012
- Predicted value Dataframe:
- Happy or Not?

```
In [201]: c = 0
happinessLevel = {}
for index, row in df.iterrows():
    if row[0] > row[1]:
        happinessLevel[index] = "Unhappy"
    else:
        happinessLevel[index] = "Happy"
        c += 1

print("Employee ID - " + test_EmpId[index] + " - " + happinessLevel[index])
print(c)
```

```
Employee ID - fffe31003300390039003000 - Unhappy
Employee ID - fffe31003300310037003800 - Happy
Employee ID - fffe33003400380035003900 - Unhappy
Employee ID - fffe3100370039003200 - Unhappy
Employee ID - fffe32003600390036003700 - Unhappy
Employee ID - fffe3600390032003200 - Happy
Employee ID - fffe3600370032003200 - Happy
Employee ID - fffe32003900390030003000 - Happy
```

	0	1
0	0.999855	0.000145
1	0.000007	0.999993
2	0.997561	0.002439
3	0.997644	0.002356
4	0.938693	0.061307
...
6483	0.943239	0.056761
6484	0.420531	0.579469
6485	0.015073	0.984927
6486	0.999822	0.000178
6487	0.981249	0.018751

6488 rows × 2 columns



RESULT

- Confusion Matrix:

Confusion matrix

```
[[2268  287]
 [ 354 3579]]
```

True Negative(TN) = 2268

False Positives(FP) = 287

False Negative(FN) = 354

True Positive(TP) = 3579

- Total Number of 'HAPPY' and 'UNHAPPY' Employees:

```
values = y_test.value_counts().keys().tolist()
counts = y_test.value_counts().tolist()
for i in range(len(values)):
    if values[i] == 0:
        print("Number of 'NOT HAPPY' Employees are: "+str(counts[i]))
    else:
        print("Number of 'HAPPY' Employees are: "+str(counts[i]))
```

Number of 'HAPPY' Employees are: 3933

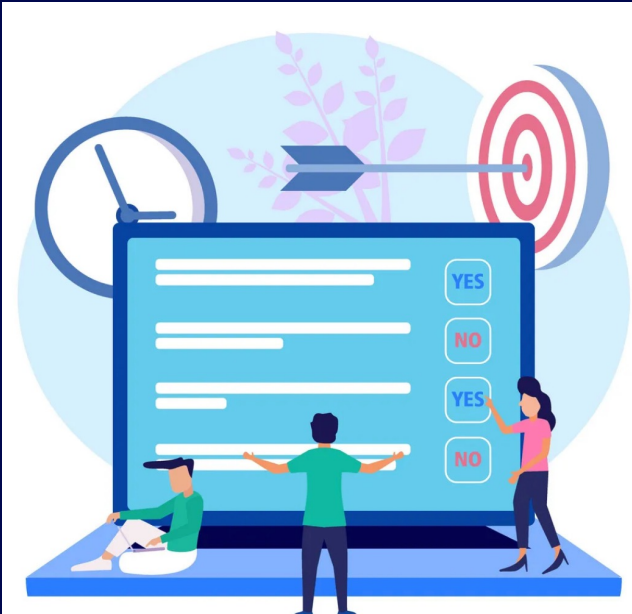
Number of 'NOT HAPPY' Employees are: 2555



DEMONSTRATION

- http://localhost:8888/notebooks/Downloads/archive/Input/AI_Project/EmpHappinessMonitoring.ipynb#

POSSIBLE EXTENSIONS



- ❑ ACCURACY - CHECK BY EMPLOYEES USING SURVEYS BASED ON OUTPUT:
 - Output can be used to generate surveys to understand whether the Machine output is accurate based on the factors.
 - Survey questions depend on the Class: Happy or Unhappy.
- ❑ DESIGN AUTOMATED BRAIN GAMES BASED ON OUTPUT:
 - If the employee is unhappy, weekly schedule short brain games in free time.
- ❑ WEB APP DESIGN TO USE MODEL:
 - Save Trained model and design Web App to upload data and predict output .



RESOURCES

- https://www.splunk.com/en_us/data-insider/ai-and-machine-learning.html
- <https://medium.com/@BonsaiAI/concept-networks-combining-subject-matter-expertise-machine-learning-to-build-industrial-ai-85dc4fa15f92>
- https://www.tutorialspoint.com/machine_learning_with_python/
- <https://www.jeremyjordan.me/evaluating-a-machine-learning-model/>
- <https://medium.com/analytics-vidhya/model-evaluation-metrics-in-machine-learning-928999fb79b2>
- <https://elitedatascience.com/model-training>
- https://en.wikipedia.org/wiki/Machine_learning



THANK YOU