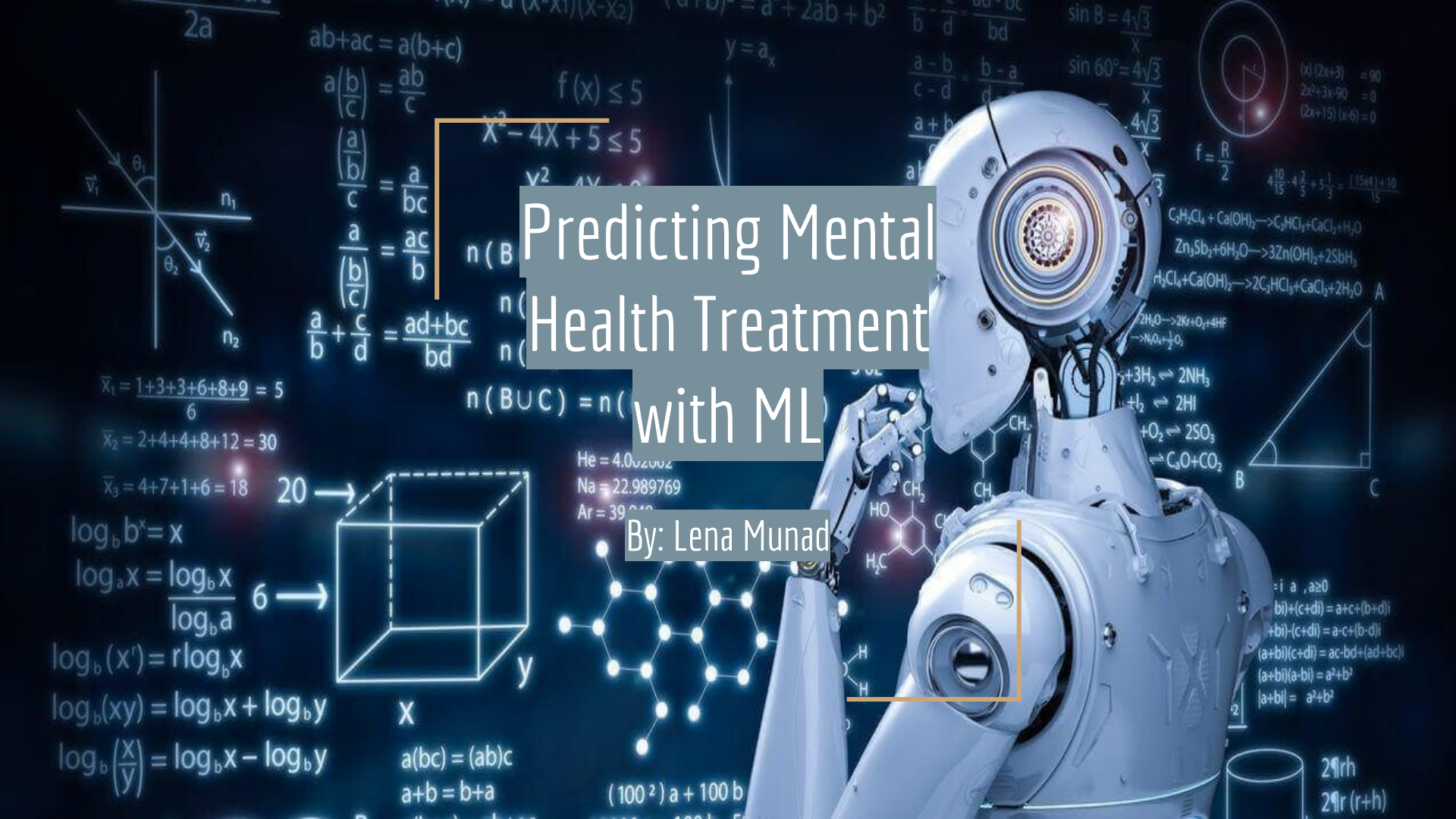


Predicting Mental Health Treatment with ML

By: Lena Munad

Introduction

Goal

- Predict if a person in tech will seek mental health treatment, based on specific factors such as remote work, family history, etc.

Importance

- Guides tech companies, specifically HR teams, in creating more supportive and inclusive environments by identifying key factors for mental health treatments.

Task type

- Binary classification using survey dataset from Kaggle

ML Models Used

- k-Nearest Neighbors (kNN) → simple, good baseline
- Naïve Bayes (Gaussian vs Categorical) → Fast, works with categorical data
- Support Vector Machine (SVM) → works well with high-dimensions

Dataset & Features

Dataset Used

- Kaggle - Mental Health in the Tech Workplace in 2014
- ~1,200 responses
- *Loaded in as survey.csv (see below)*

Features

- Overall: remote work, benefits, family history, anonymity, etc
- Target: *treatment (yes = 1, no = 0)*

[3]:

	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	...	leave	mental_hea
0	2014-08-27 11:29:31	37	Female	United States	IL	NaN	No	Yes	Often	6-25	...	Somewhat easy	
1	2014-08-27 11:29:37	44	M	United States	IN	NaN	No	No	Rarely	More than 1000	...	Don't know	
2	2014-08-27 11:29:44	32	Male	Canada	NaN	NaN	No	No	Rarely	6-25	...	Somewhat difficult	
3	2014-08-27 11:29:46	31	Male	United Kingdom	NaN	NaN	Yes	Yes	Often	26-100	...	Somewhat difficult	
4	2014-08-27 11:30:22	31	Male	United States	TX	NaN	No	No	Never	100-500	...	Don't know	

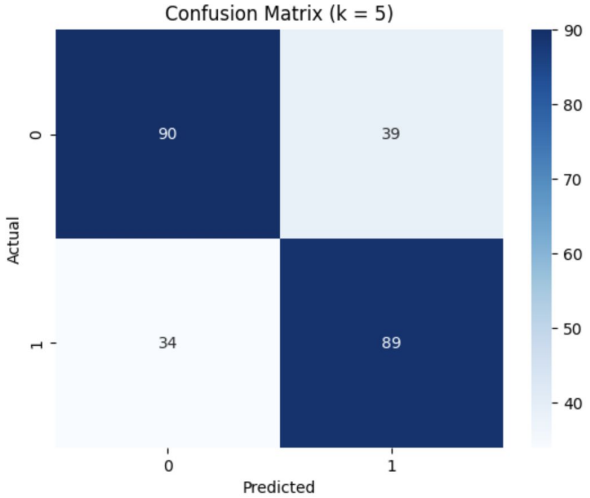
5 rows × 27 columns

k-Nearest Neighbors (kNN)

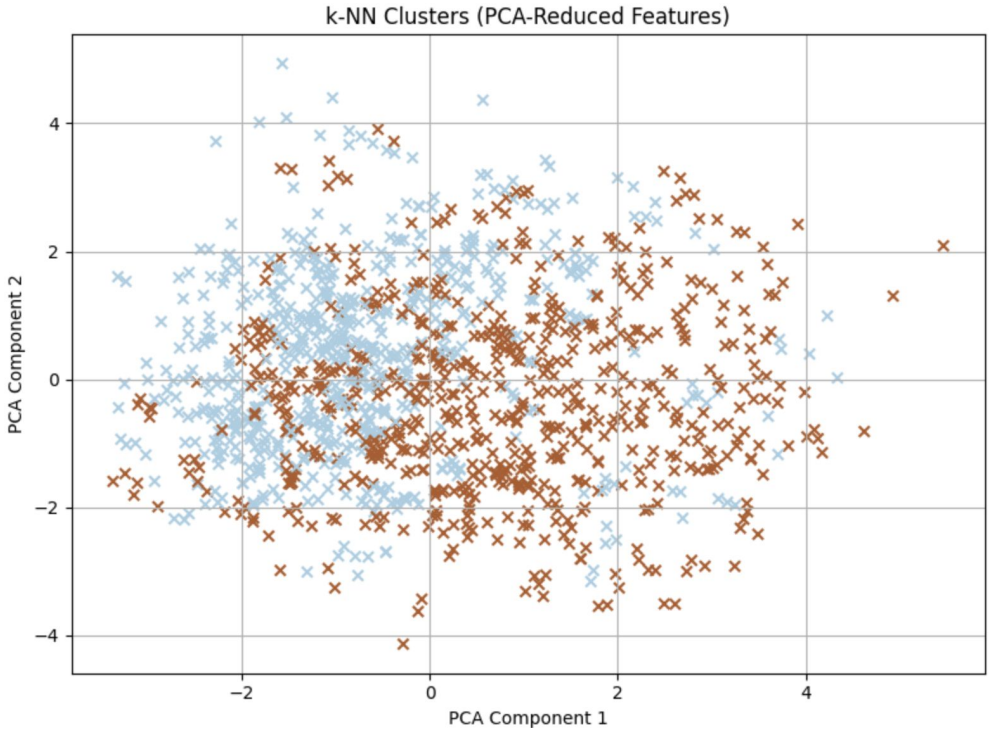
kNN Summary

- a simple, distance-based algorithm that classifies samples based on their nearest neighbors (k)
- easy to interpret, especially for small to medium datasets
- normalize data for fair distance comparisons, test different values of k for optimization
- applied PCA to reduce the feature space and plotted the predicted clusters in 2D *(as seen in next slide)*

```
k-NN Evaluation (k = 5):
Accuracy: 0.7103174603174603
Precision: 0.6953125
Recall: 0.7235772357723578
F1 Score: 0.7091633466135459
Confusion Matrix:
 [[90 39]
 [34 89]]
```



k-Nearest Neighbors (kNN)



Naïve Bayes (GaussianNB)

Naïve Bayes Summary

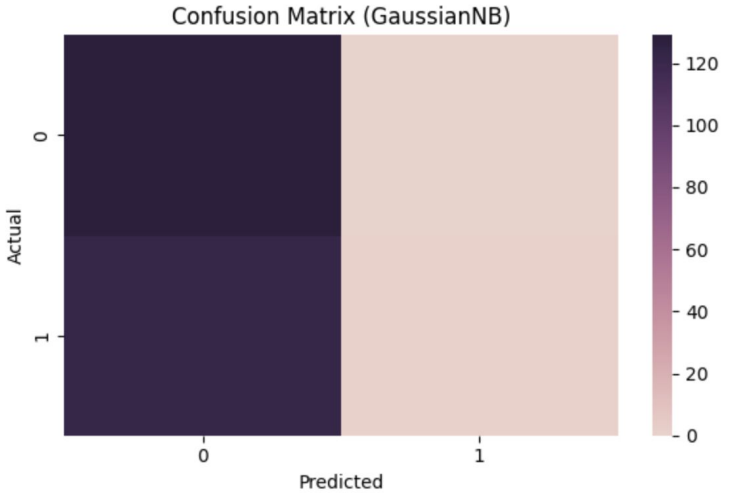
- ➢ Probabilistic classifier based on Bayes' Theorem
- ➢ Assumes features are conditionally independent given the class
- ➢ Chosen for being fast, lightweight, and effective on small to medium datasets

GaussianNB

❌ GaussianNB assumes continuous, normally distributed features (Invalid)

- ➔ (Overfitted) Confidently misclassify nearly everything (ignoring ~122 yes)!
  - ◆ Misleading statistical patterns, because invalid assumptions
- ➔ Performed EXTREMELY poor :(

```
Naïve Bayes Evaluation:
Accuracy: 0.5158730158730159
Precision: 1.0
Recall: 0.008130081300813009
F1 Score: 0.016129032258064516
Confusion Matrix:
 [[129   0]
 [122   1]]
```



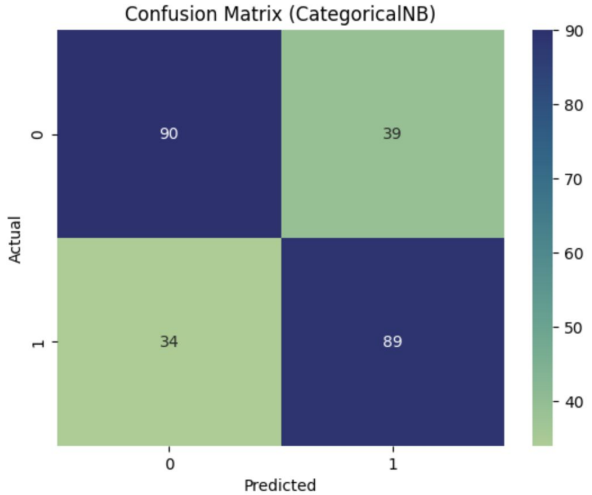
Naïve Bayes (CategoricalNB)

CategoricalNB

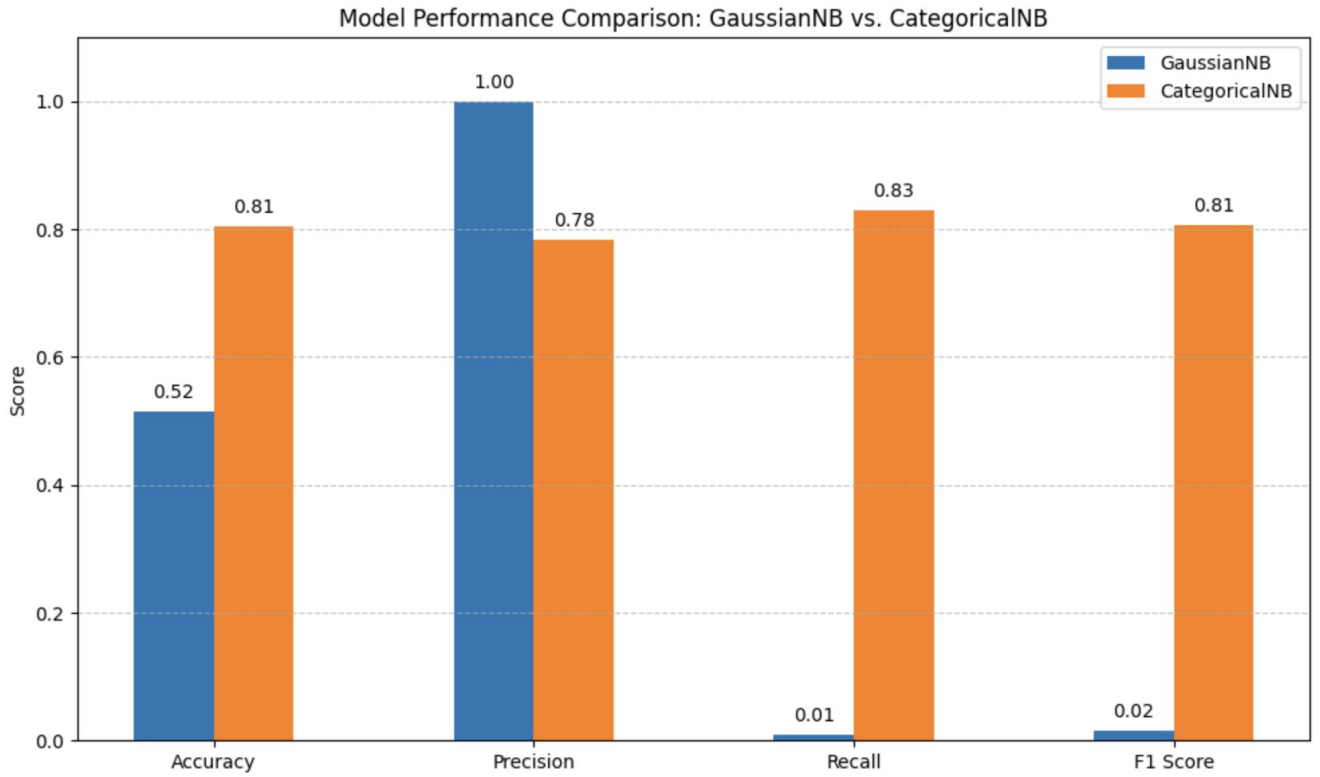
Switched to CategoricalNB because it works better with yes/no and other simple category data

- ➔ Assumptions
  - ◆ each feature is independent
  - ◆ features are categories turned into numbers (“yes” = 1, “no” = 0)
  - ◆ it learns from how often each value shows up for each class
- ➔ Used columns with a small number of options and turned words into numbers using label encoding

```
CategoricalNB Evaluation:
Accuracy: 0.8055555555555556
Precision: 0.7846153846153846
Recall: 0.8292682926829268
F1 Score: 0.8063241106719368
Confusion Matrix:
 [[101  28]
 [ 21 102]]
```



GaussianNB vs CategoricalNB

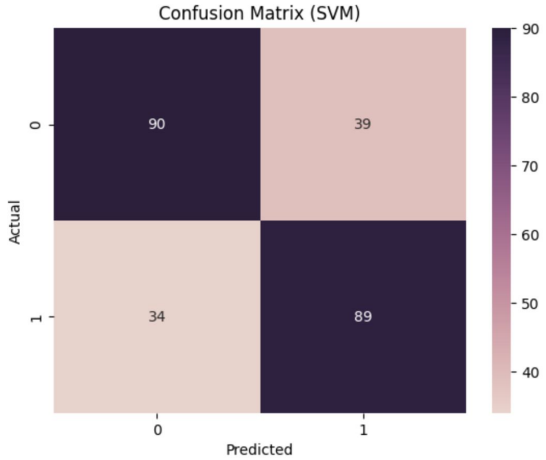


Support Vector Machine (SVM)

SVM Summary

- a classification model that finds the best boundary between two classes (yes/no)
- handles complex, non-linear patterns well using kernel functions (rbf, sigmoid, poly, etc)
- Assumptions
  - data can be separated and most important points are near the boundary (support vectors)
- tested different kernels (like RBF), tuned parameters like C and gamma, and scaled the features
- compare its results with k-NN and Naïve Bayes using accuracy, precision, recall, and F1 score

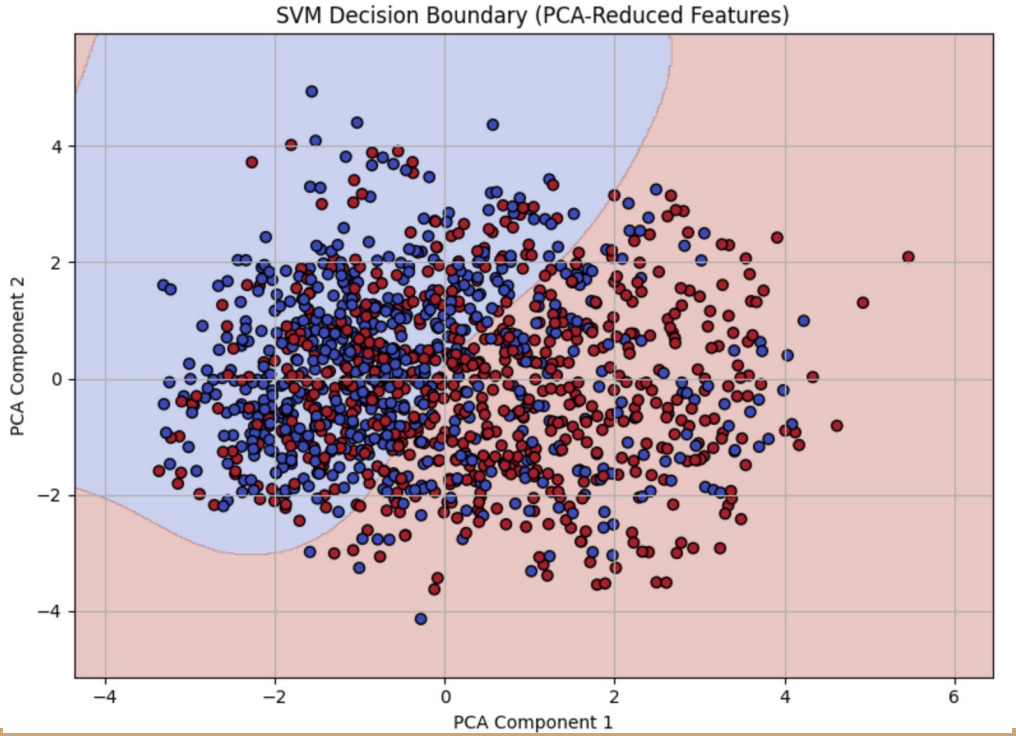
```
SVM Evaluation:
Accuracy: 0.6904761904761905
Precision: 0.6923076923076923
Recall: 0.6585365853658537
F1 Score: 0.675
Confusion Matrix:
 [[93 36]
 [42 81]]
```



kernel='sigmoid', C=1.0, gamma='scale'

Support Vector Machine (SVM) w/PCA

- applied PCA to reduce the feature space and plotted the predicted decision boundaries in 2D *(refer below)*



Conclusion & Key Takeaways

- Categorical Naïve Bayes performed best out of the 3 models
  - *Accuracy: 80.6%, F1: 80.6%*

- k-NN (k=5) was a strong baseline
  - *Accuracy: 71.0%, F1: 70.9%*

- GaussianNB overfitted (all predictions for “no treatment”) due to poor fit for categorical dataset
  - *Accuracy: 51.6%, F1: 1.6% :((*

- PCA visualizations showed clear class separation and supported model interpretation

Challenges & Future Improvement

Challenges

- Data cleaning, inconsistent categories, erasing null values
- Poor fit of GaussianNB on categorical data :(

Future Improvement

- Don’t pick Naïve Bayes (Gaussian) for categorical data!!! 😭
- Use Random Forest or Boosting
- Apply GridSearchCV for tuning
- Analyze feature importance

Github link to my code: https://github.com/IceyGirl424/CS5-machine-learning


THANK YOU