



# Kai: An AI-powered Chatbot to Support Therapy

Thesis Defence Presentation  
Mariama C. Djalo D.

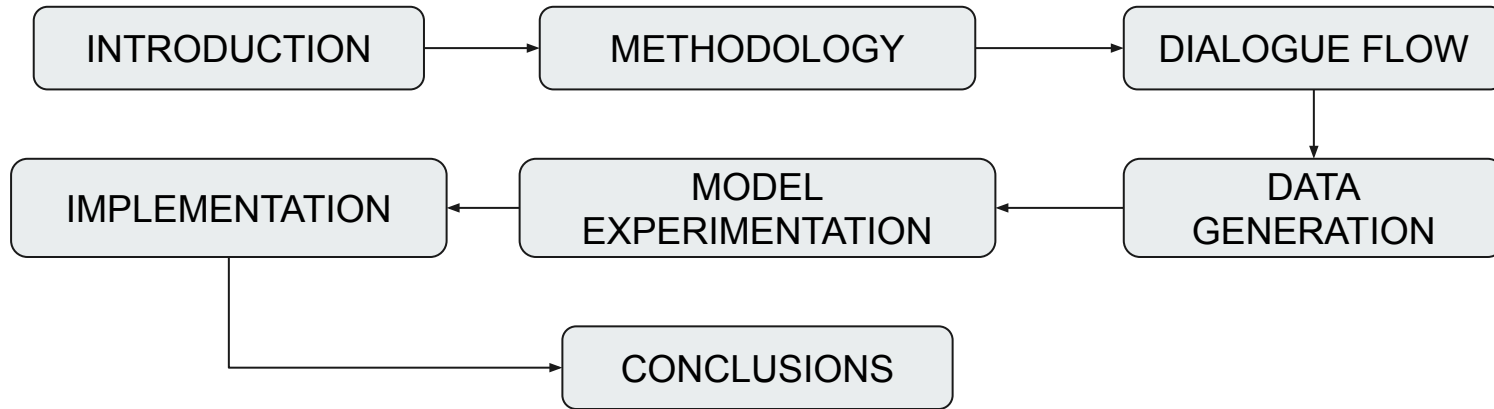


# ABSTRACT

---

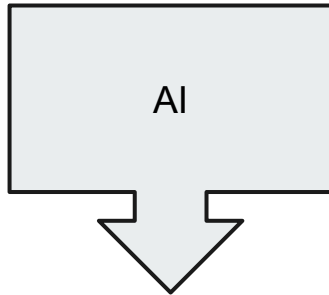


# OVERVIEW

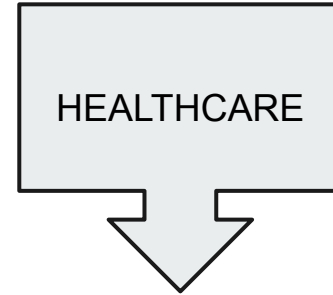




# INTRODUCTION

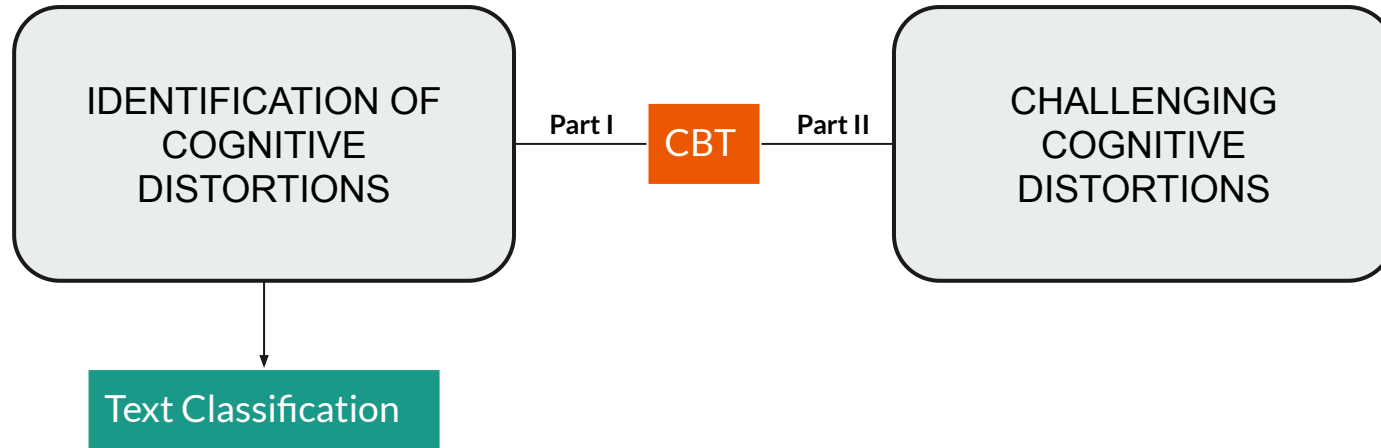


Text Classification



CBT

# INTRODUCTION

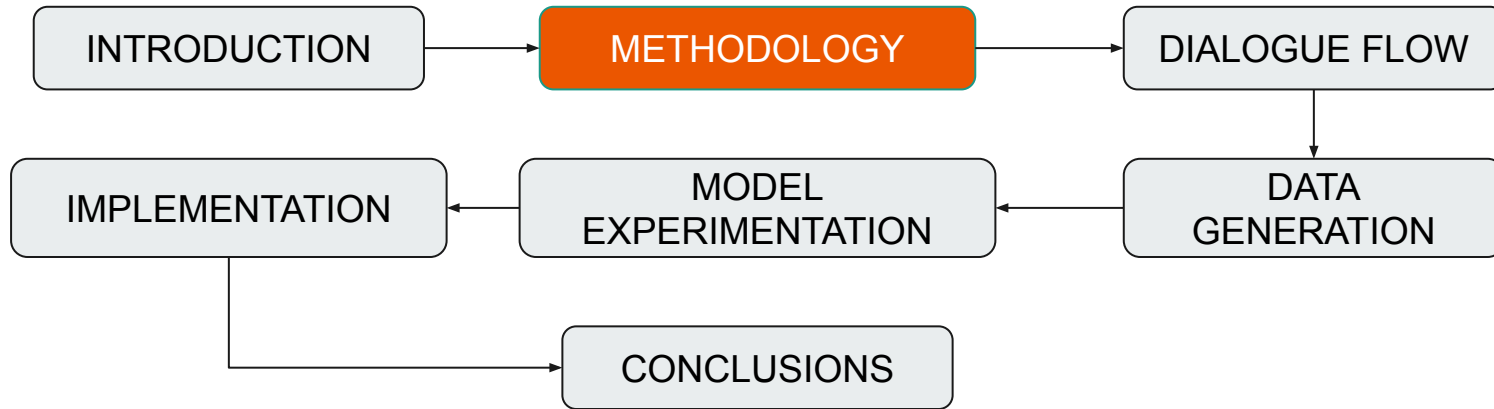




# INTRODUCTION

Cognitive Distortion	Definition	Example
<b>Jumping to Conclusions</b>	Drawing judgments without enough evidence.	<i>"She's not responding so she must be ignoring me or even worse, she must be mad at me"</i>
<b>Mental Filter</b>	Focusing only on the negative part of everything.	<i>"I did well on most of the exam, but I got one question wrong, so I must be a complete failure."</i>
<b>Emotional Reasoning</b>	Thinking that because you feel something it must be real.	<i>"I feel like a failure, so I must be a failure."</i>

# OVERVIEW





# METHODOLOGY

Dialogue  
Flow

Data  
Generation

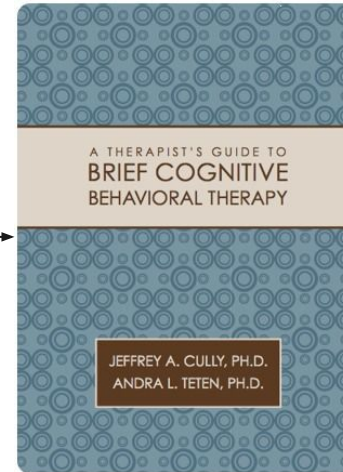
Model  
Experimentation

Implementation



# METHODOLOGY
















Dialogue  
Flow



A Therapist's Guide to Brief Cognitive Behavioral Therapy by Jeffrey A. Cully et al.

# METHODOLOGY

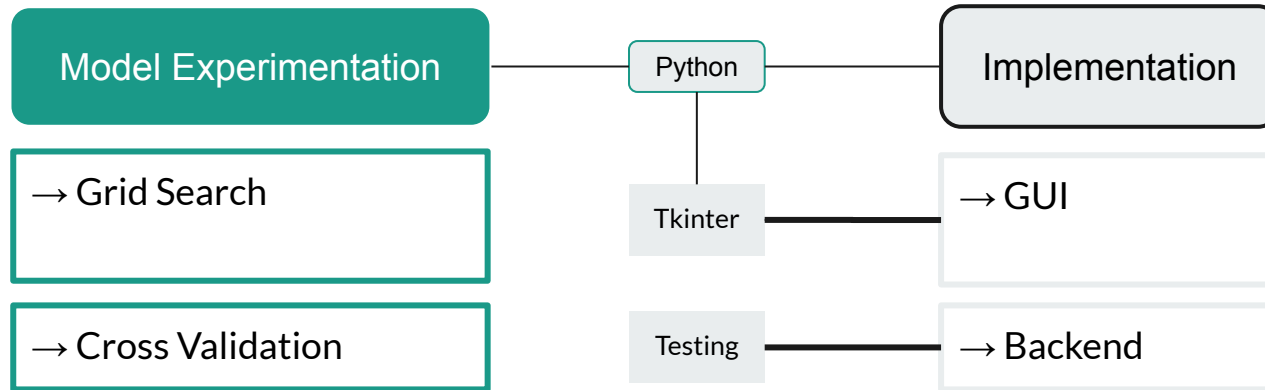
Data  
Generation

Cognitive Distortions	
 <b>FILTERING</b> Focusing on the negative Ignoring the positive	 <b>CATASTROPHIZING</b> Expecting the worst case scenario Minimizing the positive
 <b>POLARIZED THINKING</b> All or nothing thinking Ignoring complexity	 <b>HEAVEN'S REWARD FALLACY</b> Expecting self-reliance to be rewarded
 <b>CONTROL FALLACIES</b> Assumes only others to blame Assumes only self to blame	 <b>ALWAYS BEING RIGHT</b> Being wrong is unacceptable Being right excuses everything
 <b>FALLACY OF FAIRNESS</b> Assumes life should be fair	 <b>PERSONALIZATION</b> Always assuming self responsible
 <b>OVERGENERALIZATION</b> Assumes a rule from one experience	 <b>JUMPING TO CONCLUSIONS</b> Makes assumptions based on little evidence
 <b>EMOTIONAL REASONING</b> "If I feel it, it must be true"	 <b>BLAMING</b> Assumes everyone else at fault
 <b>FALLACY OF CHANGE</b> Expects others to change	 <b>GLOBAL LABELLING</b> Extreme generalization
 <b>"SHOULD"</b> Holds tight to personal rules of behaviour Judges self and others if rules broken	

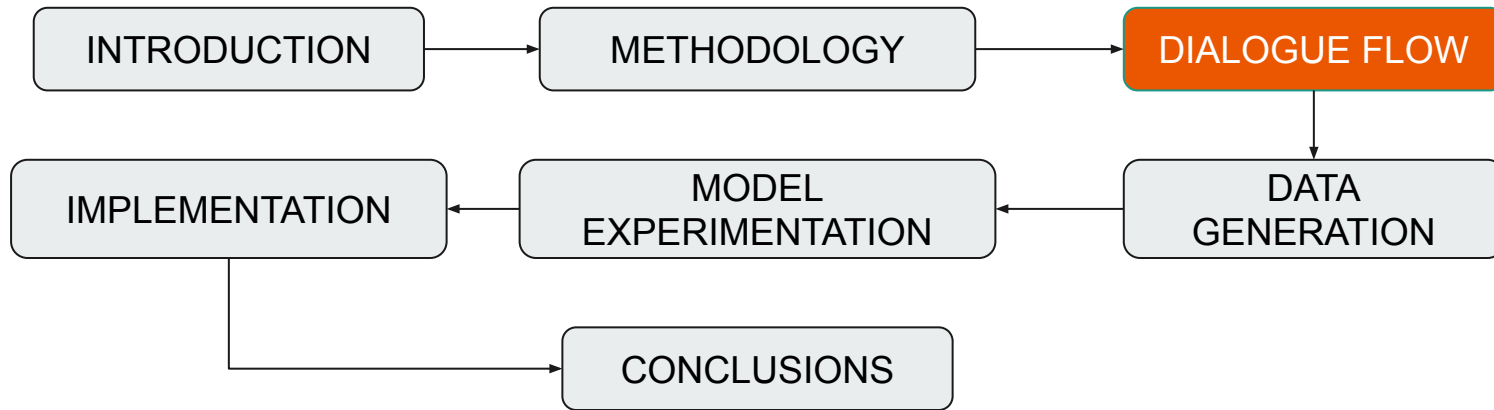
Bias?

15 Cognitive  
Distortions by  
PositivePsychology.com

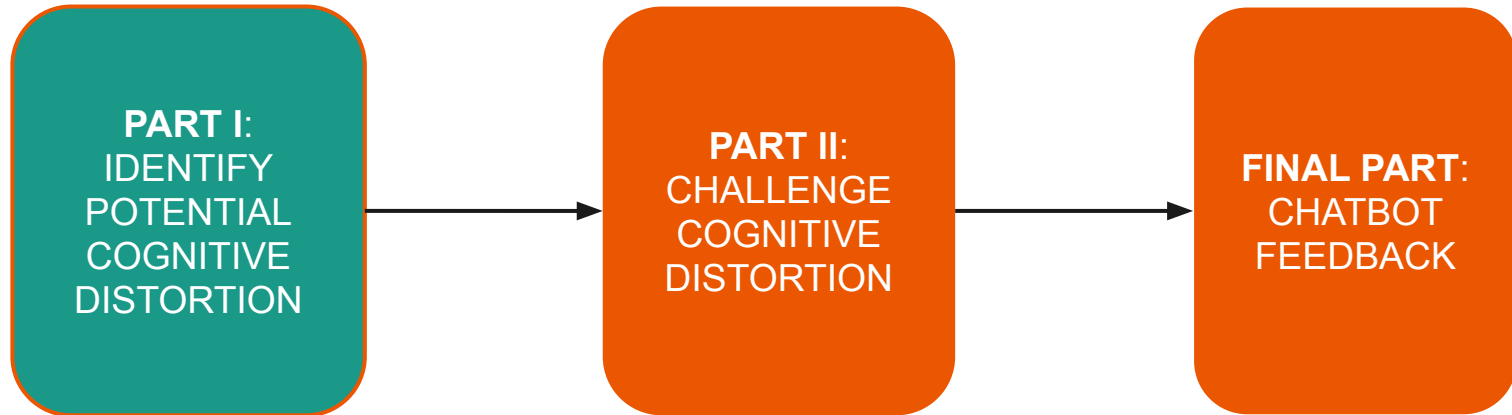
# METHODOLOGY



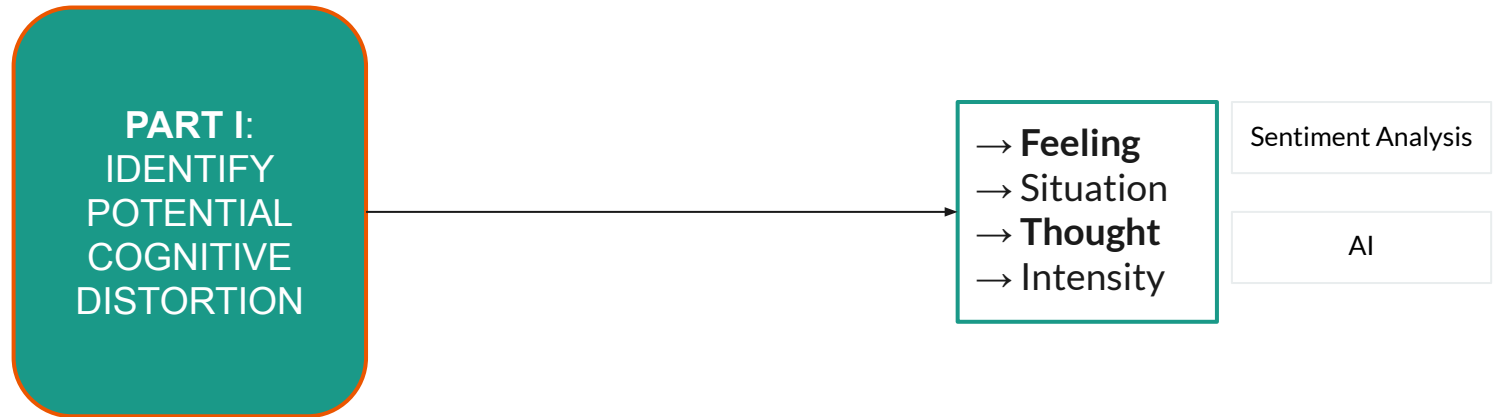
# OVERVIEW



## DIALOGUE FLOW

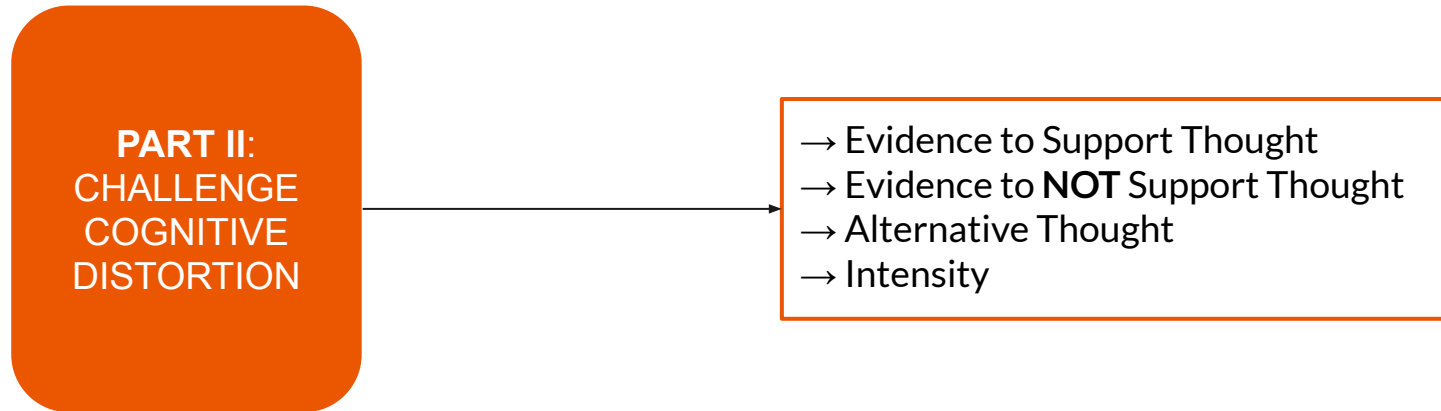


# DIALOGUE FLOW





## DIALOGUE FLOW



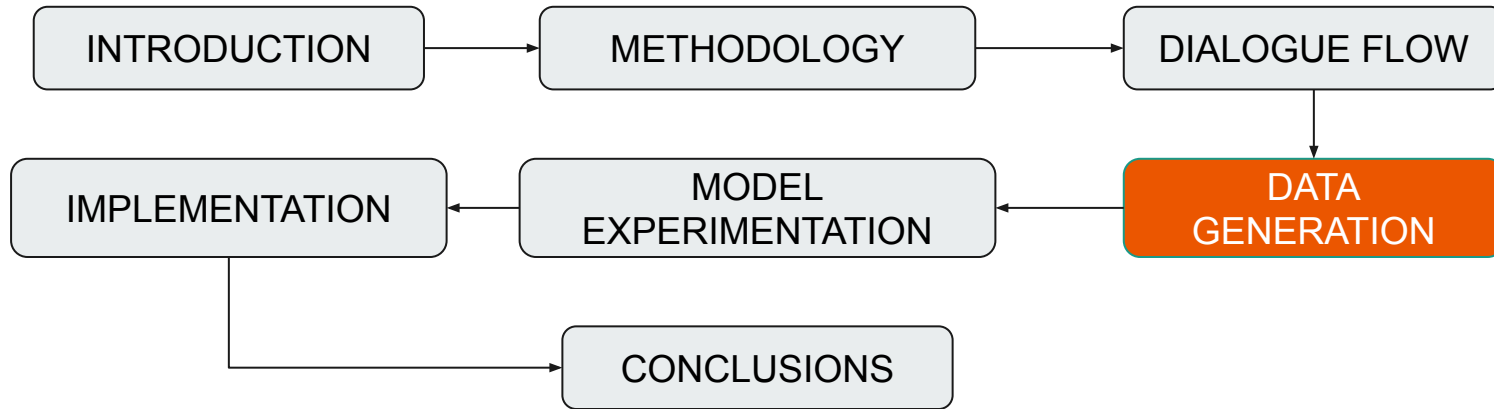
## DIALOGUE FLOW



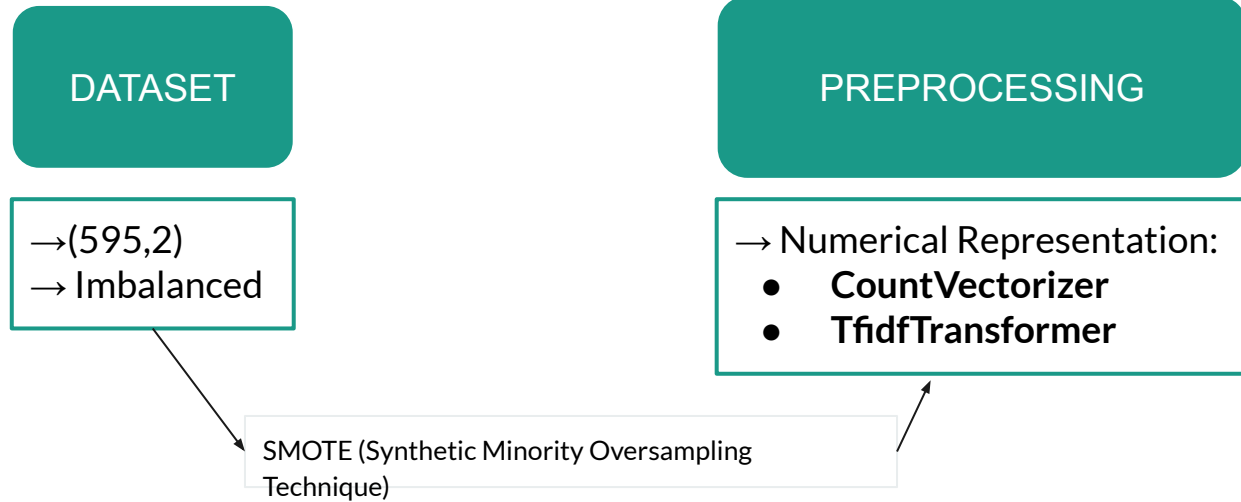




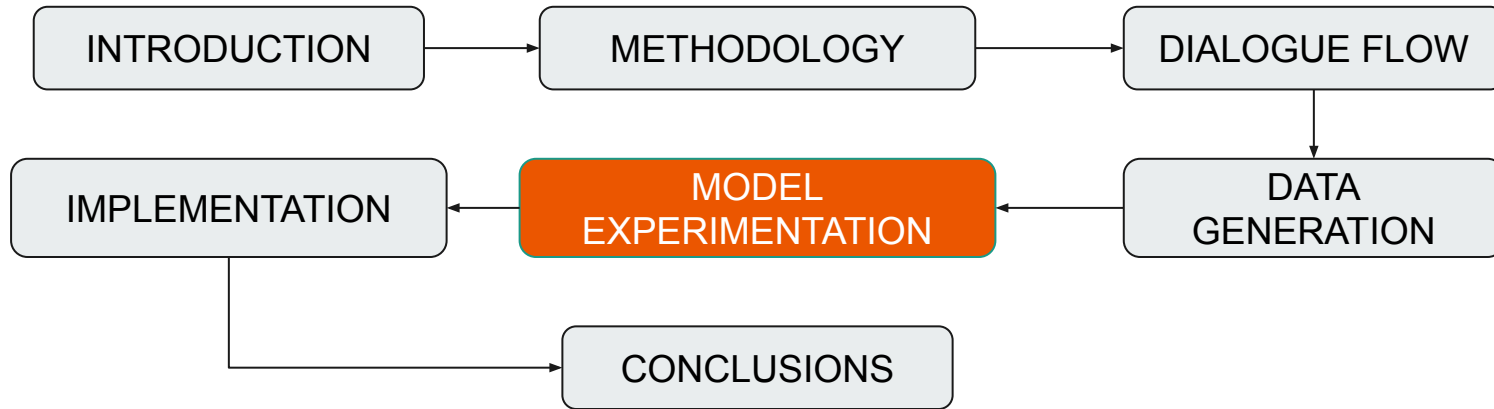
# OVERVIEW



# DATA GENERATION



# OVERVIEW





# MODEL EXPERIMENTATION

Multinomial  
Naive  
Bayes

SVM

Multinomial  
Logistic  
Regression

KNN

Random  
Forest




# MODEL EXPERIMENTATION

MODEL	BEST PARAMETERS	ACCURACY	PRECISION	RECALL	F1-SCORE	GRIDSEARCHCV
Random Forest	-	0.69	0.71	0.69	0.68	0.67
Multinomial Naive Bayes	<i>alpha: 0.1</i>	0.69	0.71	0.69	0.69	0.68
Support Vector Machine	<i>C: 1.0, decision_function_shape: ovo</i>	0.67	0.75	0.67	0.67	0.64
Multinomial Logistic Regression	<i>C: 0.01, penalty: none, solver: saga</i>	0.65	0.65	0.65	0.64	0.69
K-Nearest Neighbour	<i>p: 2, weights: distance, k_neighbors: 10</i>	0.53	0.55	0.53	0.51	0.54

Model Performance



# MODEL EXPERIMENTATION



Multinomial  
Naive  
Bayes

**VS**

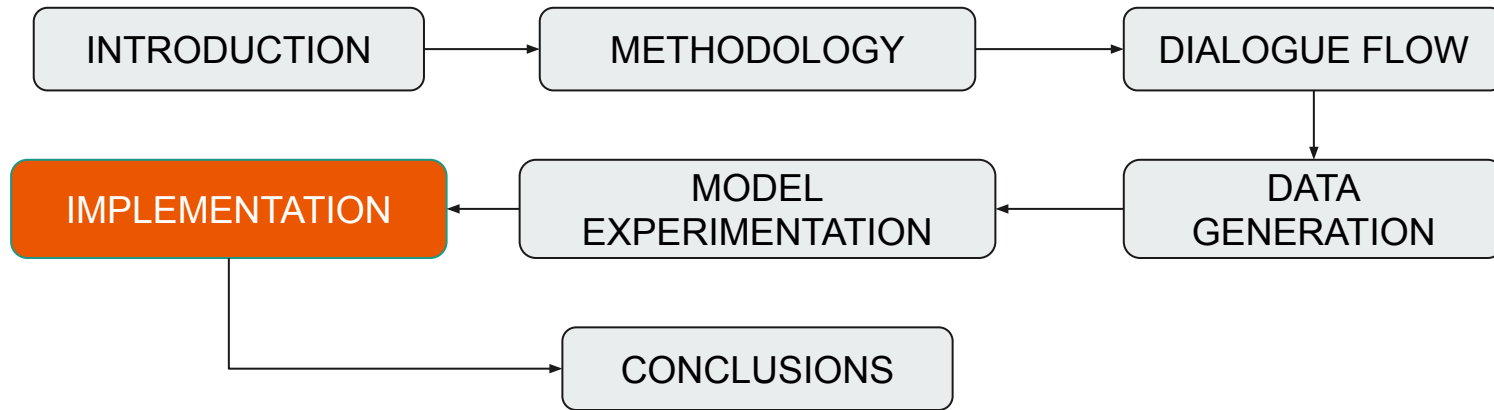


Random  
Forest

Complexity  
Feature Importance

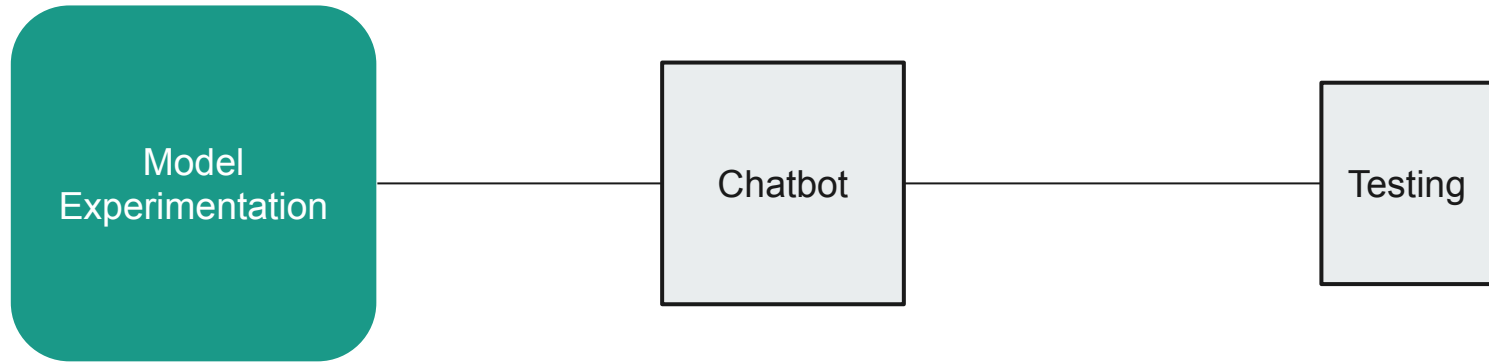


# OVERVIEW





# IMPLEMENTATION





# IMPLEMENTATION

Sklearn

```
# Divide data between training and test data
X = df['Phrase']
y = df['Cognitive Distortion']
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, train_size = .8)

MULTINOMIAL NAIVE BAYES

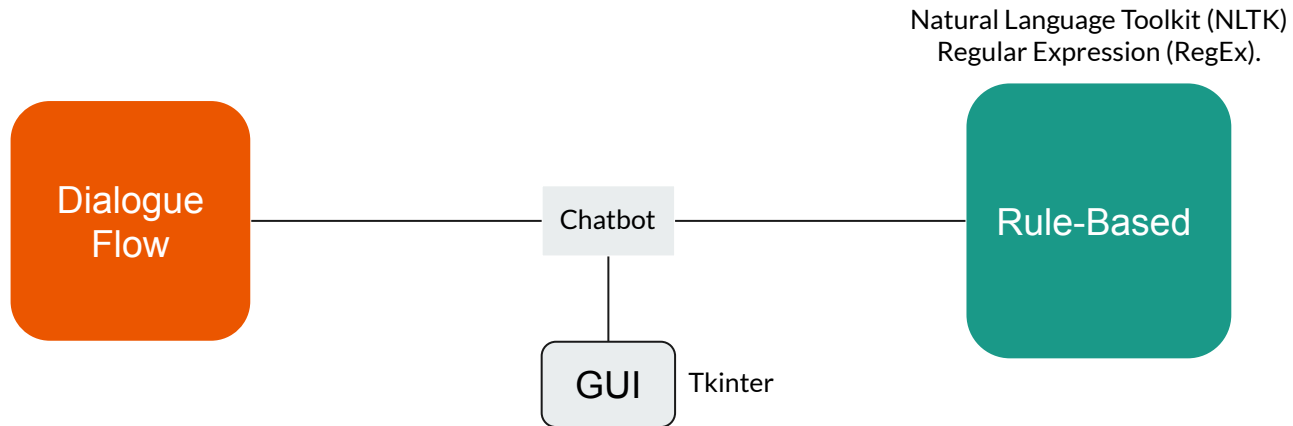
textclassifier = Pipeline([
    ('vect', CountVectorizer()),
    ('tfidf', TfidfTransformer()),
    ('smote', SMOTE(random_state=0)),
    ('mnb', MultinomialNB())
])

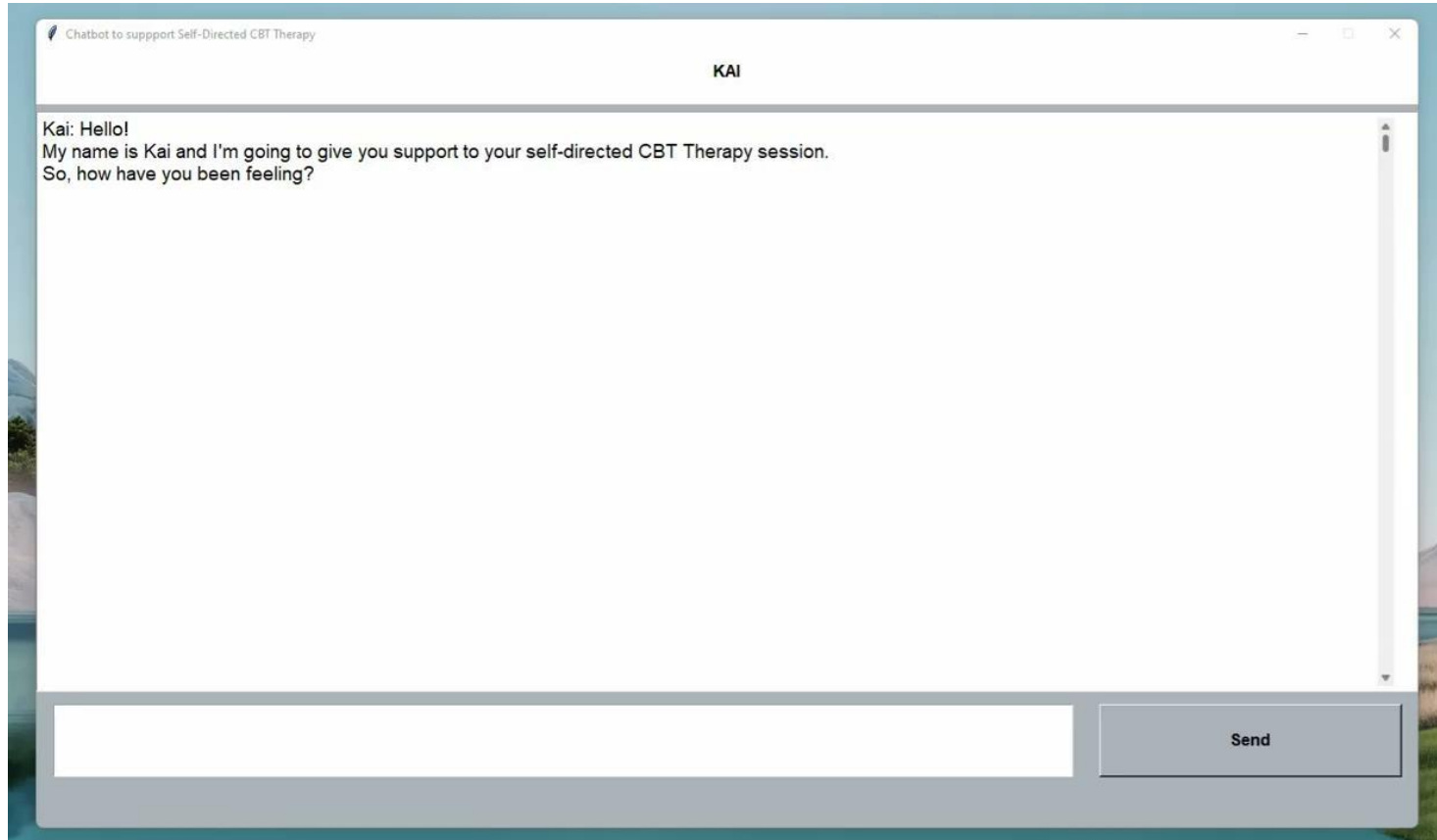
# Hyperparameters to tune
params = {'smote__k_neighbors': [2,3,4,5,6,7,8,9,10],
          'mnb__alpha': [0.01, 0.1, 0.3, 0.5, 1.0]
          }

# Hyperparameters tuning
multinomial_nb_grid = GridSearchCV(estimator=textclassifier, param_grid=params, n_jobs=10, cv=10, verbose=5)
multinomial_nb_grid.fit(X_train, y_train)
```

Model Experimentation

# IMPLEMENTATION





Demo

# CONCLUSIONS

---

**THANK YOU!**

