A. Data and Task: Data selected for the mini project is Young-People-Survey dataset and the task for this mini project is to predict how suitable a person is to work as a student volunteer to help Alzheimer's patient by predicting how empathetic he/she is.

Preprocessing and Feature Explosion of Data: Firstly, the rows containing NaN (Not a Number) were removed. Then, since the dataset included some categorical features too, it would have been wrong to drop such columns since there is no certainty of the correlation between those features and the target attribute i.e 'Empathy'. So, to retain those features, feature explosion was performed using method named "get_dummies" to convert each value of categorical feature into separate feature leading to feature explosion. Data was then splitted into train/dev/test in ratio 80:10:10.

Feature Extraction: Overall, 3 techniques were tried viz. Treebased feature selection, L1 based feature selection and PCA for dimensionality reduction. But, L1 based feature selection turned out to be best since it used L1 norm for regularization.

B. Machine Learning Solutions: Various ML classification models were implemented with the technique of parameter tuning on the development data set using GridSearch CV. Following are the models that were tried and tested to attain the maximum accuracy and to try the best of not overfitting the given data:
[The accuracy changed over different runs in the range of +/- 5%]

BASELINE MODELS:

1. **Most Frequent Strategy:** Accuracy= 29.41%
2. **Uniform Strategy:** Accuracy= 13.23%.
3. **Stratified Strategy:** Accuracy= 22.05%

ML MODELS: Each ML model did outperform the accuracy of baseline models especially after parameter tuning. Below mentioned are the accuracies after tuning.

1. **Stochastic Gradient Descent:** Accuracy= 41.17%
2. **Support Vector Classification:** Accuracy= 41.2%
3. **Logistic Regression:** Accuracy= 39.70%
4. **Random Forest:** Accuracy= 38.23%
5. **Ensemble of Classifiers:** Accuracy= 41.13%
6. **XGBoost:** Accuracy= 41.17%
7. **AdaBoost:** 35.29%

CHOSEN MODEL: Even though almost all the models have the similar accuracies, I chose SVC as my main model. Success of the model is evaluated w.r.t accuracy and not F1-score specifically.Classification report was seen to evaluate success. I chose SVM since SVM's are fast when it comes to classifying since they only need to determine which side of the "line" data lies on and they can handle complex non-linear classification. [Ensemble performs better too when used hard voting with SGD, SVM, Logistic, and Random Forest].

C. Software Used: I used PyCharm as IDE environment because, it provides the multiple interpreters for different project to work on. So, you can have two different project running on two different versions of python.

D. Examples from the Development Data set that are correctly and incorrectly classified after parameter tuning are displayed along with the results of SVC in the program. **If I were to try to fix the ones that are wrongly classified, since I am using SVC here, I would try to tune hyperparameters more , e.g. value of C is set to 100 after parameter tuning. I would have removed Gridsearch CV or would have chosen RandomSearch and set the value of C to maximum value (1000). Thus, For large values of C, the optimization will choose a smaller-margin hyperplane and if that hyperplane does a better job of getting all the training points classified correctly, it can work.**

E. References:

1. Sklearn documentation: It helped me to learn in-depth about attributes and parameters about different classification models. http://scikit-learn.org/stable supervised_learning.html#supervised-learning

2. https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/
From the above site, I learned about the XGBoost model and how it can be used after parameter tuning to increase the accuracy.

3. I went through some of the **stack overflow** forums and discussions to find the reason behind using one model over other. It was also used to rectify many errors and bugs in code.

4. **Course In Machine Learning:** I referred the textbook for the ensemble methods and bagging/ boosting classification methods.