**Final Project Report**

**Team member: Daoyuan Chen, Haopeng HU**

**Slide1**: Introduce our team name, team members, team member’s information and their picture.

**Slide2**: Describe the goal of our project, and the models which we will apply to make predictions about the potential of employees leaving the company.

**Slide3**: Before we build those models to make an prediction, at first we should have some data preprocessing with attrition\_data.csv, because this data is actual data and it’s not adaptable for all models, so before we using those data to build models, we should have some data preprocessing. It includes five steps:

1.Convert the data type of "JOBCODE" from "numeric" to "factor”

2. Replace all "" with a random ethnicity in "ETHNICITY”

3. Replace all "" with "Unknown” in "REFERRAL\_SOURCE” and remove "" level from "REFERRAL\_SOURCE”

4. Convert the data type of "REHIRE" from "logical" to "factor"

5. Use 30% test 70% training data

**Slide4**: First, we will build KNN model to make a prediction, this slide introduces the definition and the advantage of KNN Algorithm.

**Slide5**: For this slide, we build three model by KNN Algorithm for K=3, K=5 and K=10, first we will build the K=3 model, we will use kknn library and combine STATUS with other columns except EMP\_ID to build the prediction model with training data, then we table the STATUS from test data and K=3 model’s prediction and compare the prediction(k=3) with actual test data to get the accuracy, and we get the result, when k=3, the KNN model’s error\_rate is 0.3866158.

**Slide6**: This slide we will build the K=5 model, the build of this model is as same as K=3 model, then we will table the STATUS from test data and K=5 model’s prediction and compare the prediction(k=5) with actual test data to get the accuracy too, and the result of KNN model’s (K=5) error rate is 0.386369.

**Slide7**: This slide we will build the K=10 model, it is as same as K=3 model too, after building of the model we will table the STATUS from test data and K=10 model’s prediction and compare the prediction(k=10) with actual test data to get the accuracy too, and the result of KNN model’s (K=5) error rate is 0.3692788.

**Slide8**: This slide make a conclusion with the result of KNN’s three models(k=3,k=5,k=10), and according to the results of three models, we can see that the number of K has a little influence with the accuracy of KNN’s Prediction, and we can assume that when the number of K is bigger, the result of this KNN’s model is more accurate.

**Slide9**: For this slide, we will use Naive Bayes Algorithm to build the Naive Bayes model and make a prediction by using e1071 library and combine STATUS with other columns except EMP\_ID to build the prediction model with training data, then we table the STATUS from test data and Naive Bayes model’s prediction and compare the prediction with actual test data to get the accuracy, the result is that the error\_rate of Naive Bayes model is 0.350208.

**Slide10**: For this slide, we will use Decision Tree Algorithm to build models and make prediction, for the first, we are going to use CART and make a prediction by using rpart and rpart.plot library and combine STATUS with other columns except EMP\_ID to build the prediction model with training data, then as same as former, we will table the STATUS with test data and prediction, and we will get the result of the model’s accuracy, it’s 0.2676838.

**Slide11**: Then, we will still use Decision Tree Algorithm, and we will use C50 model, and as same as former, we will build C50 model by using C50 library, and combine STATUS with other columns except EMP\_ID by training data to build prediction, and table the STATUS from test data and prediction to make the accuracy visible, and then we can get the result that the error rate of C50 model is 0.2479196.

**Slide12**: For this slide, because of the limit of R language and the ANN model needs data are numeric type, so we will use python language to build this model, and before we build the ANN model by using Neural Networks, we need more data preprocessing to make the data adaptable as following:

1.Drop "EMP\_ID" and "TERMINATION\_YEAR"(decrease the useless columns)

2.Perform min-max scaling each numerical feature column to the range [0,1]

3.Perform one-hot encoding on each categorical feature column

4.Perform one-hot encoding on label column

5.Concatenate normalized numerical features and one-hot encoded categorical features into X and denote label by y

**Slide13**: For this slide, we will use use Keras to train a neural network with one 5-node hidden layer and a 1-node sigmoid output layer at first. compile the keras model and fit the keras model on the dataset(we will set 100 test and get the avenge accuracy)

**Slide14**: From the former calculation, we can Evaluate the accuracy of the prediction, and the accuracy of ANN’s prediction is 0.7392510175704956, so the error rate of this model is around 0.26074898243.

**Slide15**: For this slide, we build the confusion matrix and plot it to make ANN model’s Prediction visible

**Slide16**: This slide we are going to use Decision Tree Algorithm with Random Forest model, as same as ANN model, because of the limit of R language, we will use python to build this model and we need some data preprocessing as same as ANN model did. Then we can use RandomForestClassifier to build Random forest model.

**Slide17**: The slide we can calculate that the accuracy of Random Forest’s prediction is 0.7555478502080444, and we can get the error rate of this model is around 0.2444521498.

**Slide18**: Building confusion matrix and plot it to make Random Forest model’s Prediction visible.

**Slide19**: Showing the error rate for those models and draw a conclusion that, the Random Forest model, CART model, C50 model and Ann model which are based on Decision Tree Algorithm and neural network are more accurate than KNN models and Naive Bayes model which are based on KNN Algorithm and Naive Bayes Algorithm. Furthermore, the accuracy of KNN models maybe influenced by the number of K.

**Slide20**: Thanks for your watching.