Data mining techniques-Assignment 1-Basic

Ziyu Zhou, Luyao Xu, Congyi Dong
—Group 66—

Vrije Universiteit Amsterdam

1 Explore a small dataset

1.1 Overview of ODI dataset

In the Own Dataset Initiative(ODI) dataset there are 276 records which were filled in the form by 276 random students. There are 17 attributes contained in this dataset. One of them is the timestamp and the 16 attributes left are questions which needed students to answer by themselves. In table 1, we can see the basic characteristics of the ODI dataset. There are three attributes that we cannot distinguish the missing values. This is because no matter people leave them in blank or fill in 0 , the data would be 0. We present these kind of missing values as X in table 1.

Table 1. The overview of ODI

Attributes	Data type	Missing values	Attributes	Data type	Missing values
Programm	Letters	1	birthday	Letters	1
Machine learning	Categorical(two class)	3	neighbors in class	Letters/numerical	X
Information retrieval	Categorical(two class)	5	stand up	Categorical(two class)	12
statistics	Categorical(two class)	10	random number	Numerical	X
databases	Categorical(two class)	3	bed time	Letters/numerical	4
gender	Categorical(two class)	13	100 euro	Numerical	X
chocolate	Categorical(three class)	9	good day(1)	Letters	1
Stress	Letters/numerical	2	good day(2)	Letters	1

1.2 Pre-processing rules in this case

From the name of the attributes, we can intuitively deduce that 'stress level' is related to program, courses that students registered and self-comment('100 euro ranking'). Thus, we picked some attributes which we were interested in making 'Program', 'gender', 'machine learning', 'information retrieval', 'statistics', 'databases' ,and '100 ranking' as attributes and 'stress level' as class label.

a. Categorical data

'Gender', 'machine learning', 'information retrieval', 'statistics', 'databases' are categorical data. We use 'gender' as the example. Since it only has three kinds of answers: male, female and unknown, we assigned '0' to 'male' and '1' to female.

Then we need to do some transformations and deal with the missing values.We ignored the all missing values temporarily and picked the category with more answers and assigned the value of the winner category to missing ones.

b. Letters/Numerical data(typing in data)

'100 ranking' and 'stress level' are those kind of data which are typed in by students. For '100 ranking', we firstly dealt with the outliers: if it was less than 0, we assigned 0 to it; if it was more than the maximum, we assigned the maximum to it. Then, we used python to recognize the number answers and transform them into the type of float which can be dealt with easily. After this, we ignored the missing data to figure out the median value of the rest data, and then replaced the missing values with the median. Finally, we normalized the values to make it range from 0 to 1.

For 'stress level', our aim is to make it to be class labels. After we dealt with outliers,we assigned '0' to it if it is less or equal to 50; else, we assigned '1' to it.

c. Data of Letters(typing in data)

'Program' is the data consisting of letters that we need to pre-process. There are 20 different programs in this attribute. Since the data in this attribute is unordered, we decided to use *one-hot encoding* to convert them into numerical values. We extended 'Program' into 20 new attributes, each represent one program. If someone is in certain program, the corresponding value is 1, else 0, which means we convert a 276×1 matrix recording the information of 'program' into a 276×20 matrix to record same information with every element either '0' or '1'.

At last, we normalized all data. Then, we finished the pre-processing of ODI dataset.

1.3 Features of the data

a. Different properties of dataset

After pre-processing, we calculated some features of the data such as mean, maximum value, minimum value, standard deviation and quantiles as table 2.

Table 2. data features of picked attributes after pre-processing

	gender	machine learning	information retrieval	statistics	databases	100 ranking	stress level
mean	0.3053	0.6250	0.4222	0.9094	0.522059	0.3743	0.3537
std	0.4614	0.4850	0.4948	0.2875	0.5004	0.4054	0.3183
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00
max	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.25 quantile	0.00	0.00	0.00	1.00	0.00	1.00	0.00
0.5 quantile	0.00	1.00	0.00	1.00	1.00	0.15	0.00
0.75 quantile	1.00	1.00	1.00	1.00	1.00	0.80	1.00

We can see the ranges of each attributes from the above table.

We were quite interested in the stress levels of the students here and we calculated the median of stress level was around 0.25.

Then we focus on the distribution of the values. Thus, we drew histograms of the attributes we picked(Figure 1). We find it is interesting that students having lower stress occupy the largest proportion, which means most students have stress level under the average value(around 35). But there is a group of students whose stress level were quite high, which cannot be neglected.

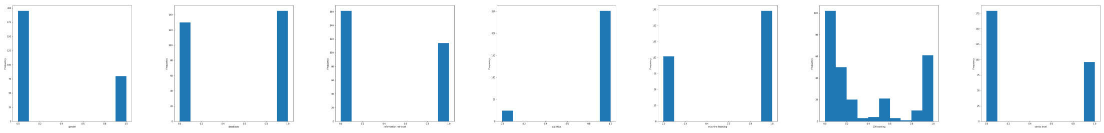

Fig. 1. The histogram of 'gender','databases','information retrieval','statistics','machinelearning','100 ranking' and 'stress level' (from left to right)

Then we picked some attributes showing relatively stronger and calculated the correlation matrix and corresponding p-values of correlation coefficients as table 3(We used IR as the abbreviation of information retrieval):

If p-value is larger than 0.05, then we can think the two variables are independent. We find it is interesting that every two attributes among the three:'machine learning','databases','information retrieval' shows strong correlation(see the bold numbers in table 3). However, other p-values are all larger than 0.05, so we think the other attributes are independent.

b. What we are interested in

Now we are curious about the distribution of female students and male students in some attributes and relationships between gender and programs.

Table 3. correlation coefficients and corresponding p-values

	gender	machine learning	IR	statistics	databases	100 ranking	stress level
gender	(1.00,0.00)	*	*	*	*	*	*
machine learning	(-0.07,0.25)	(1.00,0.00)	*	*	*	*	*
IR	(-0.08,0.19)	**(0.29,0.00)**	(1.00,0.00)	*	*	*	*
statistics	(0.00,1.00)	(0.06,0.32)	(0.05,0.41)	(1.00,0.00)	*	*	*
databases	(-0.07,0.25)	**(0.33,0.00)**	**(0.34,0.00)**	(-0.03,0.62)	(1.00,0.00)	*	*
100 ranking	(0.00,1.00)	(0.10,0.10)	(0.04,0.51)	(-0.02,0.74)	(0.07,0.25)	(1.00,0.00)	*
stress level	(0.03,0.62)	(-0.10,0.10)	**(-0.13,0.03)**	(-0.07,0.25)	(-0.04,0.51)	(-0.06,0.32)	(1.00,0.00)

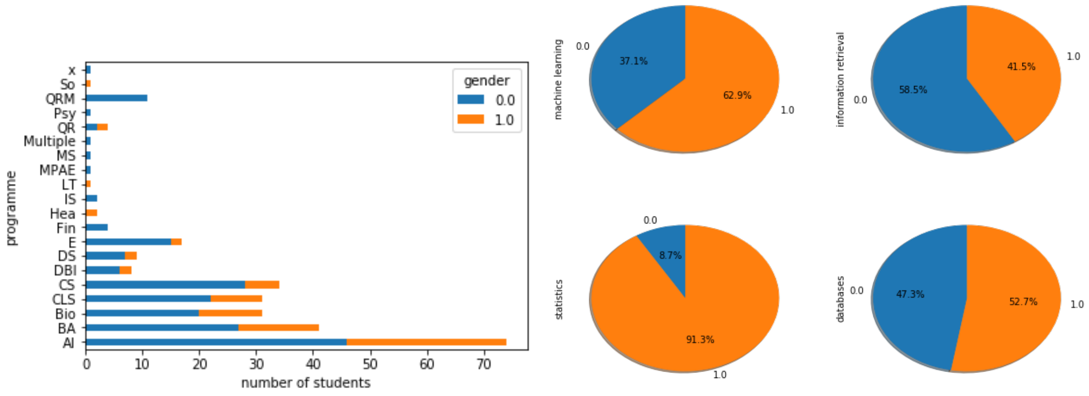

Fig. 2. Relationship between attributes and gender.

We drew a bar chart containing attributes of gender and programs(the left one of Figure 2).

We found most students filling the questionnaire were from AI. And all the students from program QRM so we made a inference that there were more male students in this program.

We also drew pie charts (the right one of Figure 2).

We found more female students having taken machine learning, statistics and databases but more male students having taken the course of information retrieval. It seems that male students were more interested in information retrieval than female students.

We are also interested in the relationship between gender and 100 ranking, also stress level. We calculated mean values of these two attributes of two genders as above.

Table 4. mean values of 100 ranking and stress level of each gender

	100 ranking	stress level
male	0.356718	0.340667
female	0.355625	0.362250

We found that the mean values are quite close, which means a balanced development of mental health between men and women.

1.4 Basic classification/regression

a. Design a classification/regression

We did the classification with 'stress level' as class label, and the rest part as sample vector.

(i)The coding steps of K-NN algorithm:

Step 1: Input the scale of neighbors K;

Step 2: Find the K neighbors samples which has the shortest distance with the test sample. In this case we decide to use Euclidean distance model to do the measurements;

Step 3: Find the class labels of those K-nearest training samples;

Step 4: Put the test sample into the class that most K-nearest samples are in, and do the cross validation to evaluate the K-NN model in this case.

(ii)The coding steps of SVM(Support vector machine) algorithm:

Step 1: Set the parameter of SVM function from sklearn. In this case, we adjust penalty parameter(depend on C) and the number of support vectors(depend on gamma);

Step 2: Figure out if the data is linear separable, and find the support vectors of two classes;

Step 4: Construct the separating hyperplane, making the sum of distances between support vectors and this hyperplane maximized;

Step 5: Use the separating hyperplane to do the classification, and do cross validation to evaluate the SVM model

b. The cross validation scores of K-NN and SVM

From the table 5 , we can see that with bigger scale we can have a stable and bigger score of cross validation, and that in this case when $K = 9$, the score is the best. Thus, we will use K-NN with $k = 9$ afterwards.

In terms of the parameter of SVM model, the bigger the C the higher the punishment. The bigger the gamma the less the number of support vectors machine. Automatically $gamma = \frac{1}{number-of-data}$, $C = 1$. From the table 6, we can see that the autometic set of SVM in sklearn yields the best cross validation score.

We can see that SVM with 'auto' parameters shows a better score than KNN with K=9. The structure of dataset is one of the reasons. The data is linear separable.

Table 5. The cross validation scores of K-NN model with different scale K when divided the dataset into 5 groups (cross validation parameter cv=5)

K=1	K=3	K=5	K=7	k=9	K=11
0.5566	0.5604	0.5932	0.6079	0.6221	0.6185

Table 6. The cross validation scores of SVM model with different C and gamma when divided the dataset into 5 groups (cross validation parameter cv=5)

	C='auto'	C=1000	C=10000
gamma='auto'	0.6509	0.6002	0.5750
gamma=1	0.6331	0.5784	0.5675
gamma=0.0001	0.6509	0.6002	0.5750
gamma=10	0.6257	0.6147	0.6146

c. Classification/regression of other models

We chose to do the data mining based on ODI dataset. After pre-processing the data of $'stresslevel(0-100)'$ which could be considered as the class attribute of the dataset, we had a two-class ODI dataset. We used 7 classifier and then did cross validation. Except K-NN and SVM, we used the other model all with their auto parameters in sklearn. The results are in table7.

Table 7. Cross validation scores of two-class classification

Logistic Regression	K-NN K=9	Gradient Boosting Classifier	Random forests	SVM	Gaussian Naive Bayes	Desition Tree
0.6366	0.6221	0.5928	0.5967	0.6509	0.3564	0.5675

From the table above, we can see that the Support vector machine classification performs the best(0.6509) while the gaussian naive bayes performs the worst(0.3564). Because after preprocessing the numerical data in this case is linear separable, SVM could be used in this case. Also, because there exists the correlation between the attributes, the naive bayes shows a relatively bad scores in validation. In addition, K-NN with scale 9 and Logistic regression also show a relatively good cross validation scores.

2 Compete in a Kaggle Competition to Predict Titanic Survival

2.1 Preparation

In this section, we competed in a Kaggle competition and completed the analysis of what sorts of people were likely to survive in Titanic disaster. The first step is to explore the data. There are 891 observations of 12 attributes in training set. Each attribute is described in the Table 2.1 below. Investigating the distribution of each attribute, we know that only 38% of total passengers were survived. Among 891 passengers, approximately 65% are male and 35% are female, however, the survival rate for males and females are 19% and 74% respectively, which implies that females have more chance to survive than males. Besides, 3rd class passengers account for 55% of the total population, while only 24% of them were survived. On the contrary, 1st class passengers constitute 24% but with survival rate of 63%. This leads to the fact that 1st class passengers are more likely to survive. In addition, the percentage of passengers embarking from S, C and Q port are 72%, 19% and 9% respectively.

Table 8. Description of attributes in the training set.

Attributes	Data type	Definition	Missing values
Name	object	passenger's name	0
Sex	object	passenger's sex	0
Ticket	object	ticket number	0
Cabin	object	cabin number	687
Embarked	object	port of embarkation	2
PassengerId	int	passenger's id	0
Pclass	int	ticket class(1 = 1st, 2 = 2nd, 3 = 3rd)	0
SibSp	int	number of siblings/spouses aboard the Titanic	0
Parch	int	number of parents/children aboard the Titanic	0
Age	float	passenger's age	177
Fare	float	price of ticket	0
Survived	float	survival (0 = no, 1 = yes)	0

To determine which attribute should be selected for classification, we plotted a correlation matrix in Figure3. It indicates that both *Fare* and *Pclass* are closely related to survival, but attributes *Fare* and *Pclass* are not independent, which leads to a decision to drop *Fare*. Besides, *SibSp* and *Parch* are statistically significantly associated, which gives us an idea to merge them into one attribute. The attribute *Age* is related to both *SibSp* and *Pclass*, hence it can not be used as a feature variable. Further analysis is given in Figure4, which shows the relationship between some attributes and target variable, *Survived*. As discussed above, *Sex* and *Pclass* are two attributes contributing to survival. Also, we find that passengers embarking from C port have the largest chance to survive compared with those embarking from S and Q ports. Furthermore, it appears that for some values of *SibSp* and *Parch*, the survival rate is statistically significant. Thus we merged the two attributes into a new feature measuring family size. *PassengerId* and *Ticket* are discarded from the data set because they are not associated with survival. Since passengers' name may contain information about their social status, a new attribute *Title* is added with six values: Master, Miss, Mr, Mrs, Officer and Royalty.

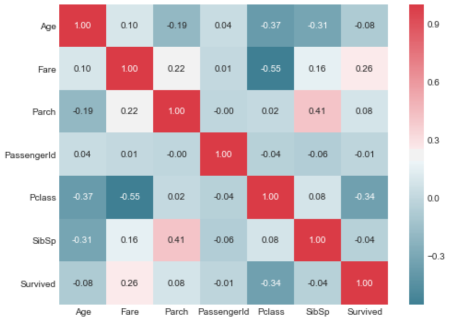

Fig. 3. The correlation matrix for attributes in training set.

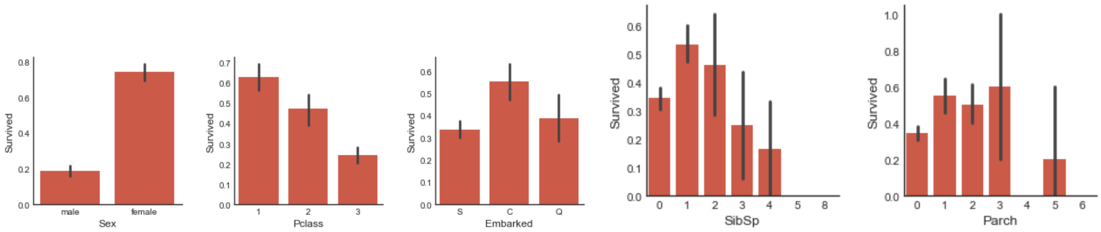

Fig. 4. Relationship between attributes and survival rate.

To process missing values in data set, we filled in missing entries of *Embarked* with 'S' because it is the most frequent entry of this attribute. However, the feature *Cabin* is discarded because it misses 77% of observations. All categorical variables are converted into indicator variables for classification. Finally, feature variables *Sex*, *Pclass*, *Embarked*, *FamilySize* and *Title* are selected and transformed to build the Titanic model.

2.2 Classification and evaluation

Two classification algorithms are used to fit the training set and make prediction based on the test set. The first training algorithm, Logic Regression, predicts the probability of occurrence of a binary event utilizing a logit function. The second algorithm used is Random Forest, which fits a number of decision tree classifiers on various sub-samples of the data set and uses averaging to improve the predictive accuracy and control over-fitting. 100 estimators are created by Random Forest classifier. Fitted models are evaluated by applying K-fold cross validation that split the training set into K (K = 10) subsamples. Each subsample is retained as validation data for assessing the model whereas the remaining subsamples are trained by the model. The average score and the standard deviation is given below (Table2.2).

Table 9. The accuracy of models evaluated by 10-fold cross validation

algorithm	mean	std	95% confidence interval
Logistic Regression	83.05%	2.91%	[77.35% , 88.75%]
Random Forest	80.48%	3.50%	[73.62% , 87.34%]

Since Logistic Regression performs better than Random Forest, this model is implemented to predict the survivors in test set. According to Kaggle's score, the accuracy of the prediction is 78.95%, which falls within the 95% confidence interval of the Logistic Regression estimator.

3 Research and theory

3.1 Research: State of the art solutions

a. The description of the competition

The competition is called Heritage Health Prize which is administered by Kaggle and sponsored by Heritage Provider Network. This competition was held from 04.04.2011 to 04.04.2013.

The data was collected during three years, including the ID, age, gender, the first claim information of 120k members.

All the teams in the competition needed to write the algorithms to predict how many days a patient would spend in the hospital the next year.

They used the degree of accuracy of their predictions of days in hospital, which was based on the RMSLE(root mean squared log error)

b. The winner and the techniques used by winner

Since the winner's paper about this competition is not available on the website, we decide to describe the method used by Thomson Van Nguyen which yielded a RMSLE score of 0.462678 that was only 0.00552 away from the winner of the competition. They used Linear regression, K-nn classifier, random decision tress to do data mining, and took the overfitting problem that would be produced in the regression analysis into consideration.

c. The main idea of the method

They used several ways to do the prediction. First, they applied the K-nn classifier by "putting people into two groups-'hospitalized'(label '1') and 'non-hospitalized'(label '0')"[1]. If people were put into 'hospitalized' group, Thomson's team then "used least squared regression model to build a weighted linear model to predict the number of hospitalization days."[1] Finally, In order to mitigate the overfitting produced by the above regression analysis through full recursion partitioning, Thomson and his team created "multiple random decision trees and combine the average score of all trees created to generate a prediction."[1]

In order to reduce the negative effects produced by overfitting, they used feature selection to "run the models with a subset of all predictors, selecting only predictors that best explain the data"[1], and repeated cross-validation.

d. What makes it stand out?

First, while the standard method only includes one approach to predict the data, in this case, Thomas used three methods together, improving the accuracy of classification. Although this methods produced overfitting, it was controlled and reduced by random decision tree and feature selection, which produced the better results when worked together than when worked alone. As the result, the winning method did improve the accuracy but not at the cost of overfitting.

3.2 Theory: MSE verse MAE

a. Corresponding formula

Set $\{Y_i\}_{i\in[1,n]}$ n predictions generated from a sample n data points and $\hat{Y}_{i i\in[1,n]}$ the corresponding observed values. Then the formula of MSE is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

Set $\{h(X_i)\}_{i\in[1,n]}$ n predictions generated from a sample n data points and $\{y_i\}_{i\in[1,n]}$ the corresponding observed values. Then the formula of MAE is as follows:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|h(X_i) - y_i| = \frac{1}{m}\sum_{i=1}^{m}|e_i| \tag{2}$$

MAE means the sum of the absolute errors.

b. Why would someone use one and not the other?

In general, MSE is more simple to calculate than MAE. However, MAE has better robustness against outliers. Since MSE takes the square of the errors, if $e > 1$, MSE will make the error even bigger, which means if the training data is contaminated by outliers, then it is better to use MAE.

To be more intuitive, if we want to get a prediction by minimizing the MSE, then the value must be the mean of all observations. But if we want to get a prediction by minimizing the MAE, then the value will be the median. As we all know, for outliers, median has better robustness than mean, which means MAE is more stable than MSE against outliers.

c. Example situation

The choice of MAE or MSE is related to the mean, the median and the errors of the sample.If all the errors are around 1, then using MSE can give us the same result as using MAE. Also, if the median and the mean of the observations are very close to each other, we can get identical results.

Let's consider this situation:

We want to estimate $\theta = h(X)$ with N observation samples $\{X_i\}_{i\in[1,N]}$. Then, based on $h(X_i)_{i\in[1,N]}$, the estimator of θ could be constructed, where $h(x)$ is a continuous function.

Supposed that the observations $h(X_i)_{i\in[1,N]}$ has the very similar mean and median, we can have the estimator $\hat{\theta}$ of θ based on MSE and MAE:

(1).Firstly, we consider the situation where we use MSE to determine the estimator. Obviously we want $MSE = \frac{1}{N}\sum_{i=1}^{N}(h(X_i) - \hat{\theta})^2$ as small as possible so the average value of observations $h(X_i)_{i\in[1,N]}$ would be the determination of estimator. Therefore, $\hat{\theta} = \frac{1}{N}\sum_{i=1}^{N}h(X_i)$ becomes the estimator of θ.

(2).Secondly, we consider the situation where we use MAE to determine the estimator. If we want MAE as small as possible, the median $\tilde{\theta}$ would be the best choice.

Since the median and the mean are very close to each other, we can get almost identical results.

d.Regression experiment

Table 10. MAE and MSE of different regression methods

	MAE	MSE
Linear Regression	0.5866	0.6667
Polynomial Regression	0.5844	0.6480
Ridge Regression	0.5864	0.6674
Lasso Regression	0.7046	0.9987

Table 11. MAE and MSE of different regression methods with more outliers

	MAE	MSE
Linear Regression	0.5866	0.6667
Polynomial Regression	0.5844	0.6480
Ridge Regression	0.5864	0.6674
Lasso Regression	0.7046	0.9987

The enterprise daily return data contains the daily log return of two companies which show relationship with each other. We obtained the data from the course of financial econometrics in UvA. To study the relationship of the daily return of the two companies, we did regression based on the data. We used four methods of regression: 1.Linear regression 2.Polynomial regression 3.Ridge Regression 4.Lasso Regression. And we got the MAE and MSE in table 10

Then we analyze the data contaminated by outliers, we changed values of random points of the data and got the results shown in table 11.

We can see from the tables that if there exists some outliers, the MSE will be much larger than MAE and MAE shows better robustness than MSE, which is consistent with the above theory and statements.

3.3 Theory: Analyze a less obvious dataset

a. Modelling techniques

For a start, we need to transform the pure text data into numerical data, by giving each text a corresponding numerical vector. Then, convert $'ham'$ into 0 while $'spam'$ into 1, making it a two-class dataset. Finally, we should train 7 different two-class classifier on the training set and do the cross validation to obtain the validation score of each classifier.

b. data transform method

Table 12. Example of data transformation

Text\Feature label	'nb'	'movie'
a nb movie	0.65	0.75
nb! nb! movie	0.40	0.90

We filtered the stop words, e.g. 'I', and punctuation to improve the quality of data. Then we set the left words as feature labels. Based on TD-IDF(term frequencyinverse document frequency), all the texts could be converted into the numerical value vector with respect to the feature labels. Here is an example shown by table 12. For the first text we convert it into numerical vector [0.65 0.75], where 0.65 and 0.75 are the TD-IDF values of words 'nb' and 'movie' respectively in the first text.

c. Cross validation scores of all classifiers and potential improvements

We can see from the table 13 that all the classifier show the good quality, as the scores are all above 0.8500. In particular, random forests shows the best scores(0.9744), which means in this case, it is the best classifier among the 7 classifiers. In order to improve the quality of classification, we could build a classifier combination with several base classifiers. The classification result of each base classifiers would be considered as a vote. The test data will be classified into a attribute with highest votes.

Table 13. The classifiers with auto params in sklearn and corresponding validation scores

Logistic Regression	K-NN	Gradient Boosting Classifier	Random forests	SVM	Gaussian Naive Bayes	Desition Tree
0.95146	0.8970	0.9601	0.9744	0.8574	0.8748	0.9683

References

1. Thomson Van Nguyen, Bhubaneswar Mishra: *Modeling Hospitalization Outcomes with Random Decision Trees and Bayesian Feature Selection*