The Prediction of Booking Destination On Airbnb Dataset

Ke Zhang
Computer Science and Engineering
University of California, San Diego
La Jolla, California
Email: kez052@eng.ucsd.edu

Zhengren Pan
Electrical and Computer Engineering
University of California, San Diego
La Jolla, California
Email: zhp007@eng.ucsd.edu

Sichao Shi
Computer Science and Engineering
University of California, San Diego
La Jolla, California
Email:sis031@eng.ucsd.edu

***Abstract*—This report is about analysis of the Airbnb dataset and the model we built to do the prediction task on the dataset. The dataset comes from an ongoing kaggle competition supported by Airbnb. We first did some comprehensive analysis on the dataset, explored most features and collected all features we thought was useful. Then we described and interpreted the prediction task and the evaluation method. During the model building process, we first referred to some methods from the winner of a similar kaggle competition —- APPC. Then we built a reasonable model for this prediction task. To predict accurately, we built a two-level classification model. The first level is a binary classifier with Voting Mechanism combining linear, logistic and polynomial regression. The second level is a multi-class classifier which is the combination of SVM and multi-class one-against-rest logistic classification. This process also included baseline description, feature selection and representation, model selection, reasoning and description, and parameter tuning. Next we describe some literature in related area we referred to. Finally, we presented our conclusion and the Kaggle competition result before the time we submitted.**

***Keywords*—Airbnb, Prediction, Kaggle, Two-level classification Model, Binary Classification, Multi-class Classification, Voting Mechanism**

I. Dataset

A. Description of the Dataset

The dataset we are researching is provided by Airbnb which contains a list of users along with their demographics, web session records, and some summary statistics. The whole dataset contains 5 csv files:train-users, test-users, sessions, countries, age-gender-bkts.

1) train-users and test-users: The train-users files contains 171239 training examples with 16 properties:

- id
- date-account-created
- date-first-booking
- gender
- age
- signup-method
- signup-flow
- language
- affiliate-channel
- affiliate-provider
- first-affiliate-tracked
- signup-app
- first-device-type
- first-browser
- country-destination
- time-stamp-first-active

The test-users have 43673 items and 15 properties.The values of country-destination are missing and that is the value we are asked to predict.

The training and test sets are split by dates. In the test set, we are expected to predict coountry destination of all the new users with first activities after 4/1/2014.

2) sessions: The sessions file is the web sessions log records for users.

The sessions file contains 5600850 examples and 6 properties: user-id, action, action-type, action-detail, device-type, secs-elapsed.There are actually 74610 different users in the file.

3) countries: The countries file contains statistics of destination countries in this dataset and their geometric information. It has information for 10 countries and their 7 different properties , such as longitude and latitude.

4) age-gender-bkts: This file contains statistics of users' age group, gender, country of destination. It consists 420 examples and 5 properties.

B. Exploratory Analysis of the Dataset

1) users' language: The language spoken is distributed as Fig.1.

It is not surprising that most users speaks English since Airbnb is a company located in US and its customers are mostly Americans.

2) users' age: The age distribution is shown as Fig.2.

In the figure, we can see that users' age are most between 24 and 36.Young users are dominant.

Value	Count	Percent
en	165,648	96.735%
zh	1,191	0.696%
fr	957	0.559%
es	753	0.44%
de	619	0.361%
ko	474	0.277%

Fig. 1: distribution of users' speaking language

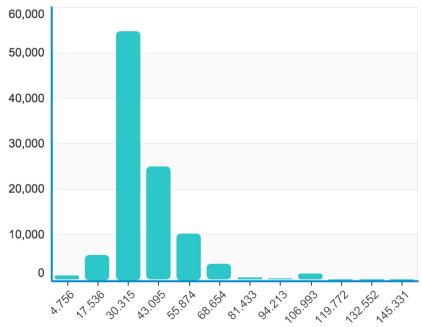
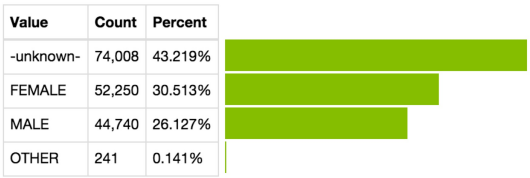


Fig. 2: distribution of users' age

3) users' gender: The gender distribution is shown as Fig.3

We can see that there are a lot of missing values for gender. Almost half of the users did not input there gender information.

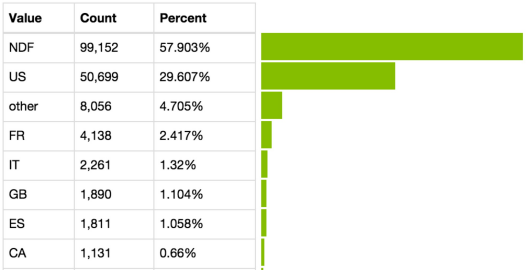


Value	Count	Percent
-unknown-	74,008	43.219%
FEMALE	52,250	30.513%
MALE	44,740	26.127%
OTHER	241	0.141%

Fig. 3: distribution of users' gender

4) users' country destination: The distribution of destination is shown as Fig.4.

We can see from the Fig.4 that most people ended up booking nothing which is indicated as NDF.Among the users who have booked in the Airbnb, US is the most popular choice.



Value	Count	Percent
NDF	99,152	57.903%
US	50,699	29.607%
other	8,056	4.705%
FR	4,138	2.417%
IT	2,261	1.32%
GB	1,890	1.104%
ES	1,811	1.058%
CA	1,131	0.66%

Fig. 4: distribution of users' destination

5) Analysis of age-gender-bkts: We computed to population ratio according to their country destination: whether it is US or not. And then sort the ratio in ascending order. Table.I is our result table.

From Table.I, we noticed that people are younger when their country destination is US while people are older when their country destination is not US.

age(US)	ratio(US)	age(not US)	ratio(not US)
100+	0.000227	100+	0.000248
95-99	0.001464	95-99	0.001491
90-94	0.00533	90-94	0.007904
85-89	0.01199	85-89	0.017352
80-84	0.01794	80-84	0.02763
75-79	0.02513	75-79	0.03816
70-74	0.03523	70-74	0.0438
65-69	0.04931	10-14	0.05191
60-64	0.05911	5-9	0.05318
40-44	0.0629	0-4	0.05330
35-39	0.06360	15-19	0.0534
5-9	0.06407	65-69	0.05443
0-4	0.06487	20-24	0.0575
45-49	0.064936	60-64	0.0595
10-14	0.064949	25-29	0.0622
15-19	0.066419	30-34	0.0653
30-34	0.06719	35-39	0.0653
55-59	0.06752	55-59	0.0662
25-29	0.06888	40-44	0.0705
50-54	0.06897	50-54	0.07516
20-24	0.06980	45-49	0.07556

TABLE I: population ratio with respect to age range

II. Predicting Task

A. Description of Predicting Task

The prediction task is to predict in which country a new user will make his or her first booking.

B. Validation of the prediction

To validate our model, we use 10-fold cross validation. In this way, each data can be used both to train and validate the model.

C. Data Pre-Processing

1) date-first-booking: To predict the country destination, in the first place we want to classify the users who booked in the airbnb and the users who did not. To do this we spit the users into two groups according to their property of country destination.In this way we found out a useful feature:date-first-booking.

There are actually 99152 users who did not booked in the Airbnb.And all of them don't have the record of date first booking. We then turned to the users who booked to ensure the effectiveness of the feature and we found out all the users

who had booked have a record of date first booking. Therefore, we can successfully predict whether a user booked or not.

Then we focused on the user who booked in the Airbnb. Since most of the country destination is US, we then want to find out a way to classify the US and Non-Us label.

Value	Count	Percent
''	99,152	100%

Fig. 5: distribution of users' date first booking who did not booked

2) The distribution of number of booking versus date: We want to find out if there is some certain distribution over the booking number and we find out that booking is concentrated in certain range. We can see it from the Fig.6.

We assume date-first-booking may be a useful feature for us to separate the users booked US and Non-US.Driven by this assumption, we plot the distribution for US users and Non-US users with different colors. In Fig.7, the X-axis denotes the time line and Y-axis denotes the number of booking while red represents US-users and blue represents Non-US-users.

According to Fig.6, spots become denser as time goes by which is reasonable because as Airbnb becomes popular, users become larger and more active. However, interestingly spots become sparse after certain date. We assume that Airbnb extract this part of the users so that they can used them as test set.

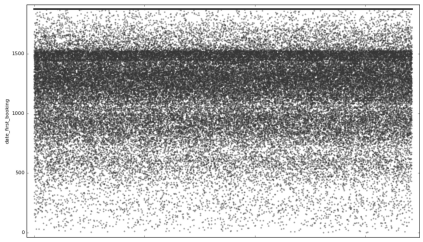


Fig. 6: distribution of number of booking versus date

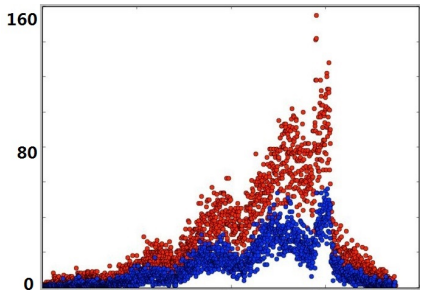


Fig. 7: distribution of number of booking versus date from two different kind of users

3) The ratio of number of booking versus month: In the Fig.8, the X-axis denotes months and Y-axis denotes the ratio of number of booking while red represents US-users and blue represents Non-US-users. Since the plotting line is different, we assume the booking month may be a useful feature.

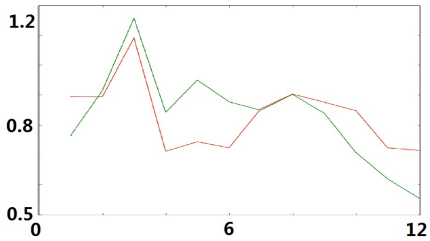


Fig. 8: The ratio of number of booking versus month

4) The ratio of number of users versus difference between booking date and account creation date: we can see from the Fig.9 that if the difference between booking date and account creation date is less than 2, namely, 0 or 1,US-users tend to have larger ratio than Non-US-users.

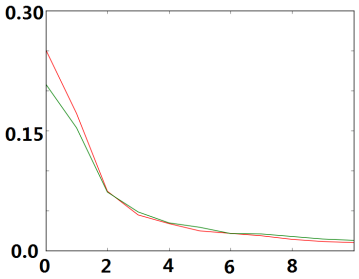


Fig. 9: The ratio of number of users versus difference between booking date and account creation date

In this case, we came up with a useful feature that if difference between booking date and account creation date is less than 2,the feature vector is $[1, 0]$,otherwise $[0, 1]$

5) age: In the Table.I, we have shown that US-users and Non-US-users tend to have different age distribution. So we come up with a age feature vector. Moreover, there are some outliers in the age set. For example, some users input 2014 as age instead of 1 years-old.(It's a actual weird case, because this user doesn't only enter the wrong number of age, but also appears too young to book online.) To handle this case, we preprocessed data to discard users with age over 100, or under 5.

D. Evaluation of Model

The evaluation metric for this prediction task is NDCG@k (Normalized discounted cumulative gain) where $k = 5$

The nDCG calculation is shown as:

$$nDCG_k = \frac{DCG_k}{IDCG_k}$$

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

where rel_i is the relevance of the result at position i

$IDCG_k$ is the maximum possible (ideal) DCG for a given set of queries. All $nDCG$ calculations are relative values on the interval 0.0 to 1.0. this validation method is that for every user's prediction, we can list at most 5 countries in order. We compare them with the validation set in this order. That is, compare the first country with the country in the validation set, if they are the same, calculate the score(in this case, it is 1) and return. There is no need to consider the rest countries. If they are different, then compare the second country with that in the validation set, if same, calculate the score and return, otherwise turn to the third, fourth, and even the last country to compare. The position of the country in the list indicates the confidence we give to that country. The more front the position of country is in the list, the more likely we think it is the result. And the score of one prediction is decided by the position of the right result, if the first prediction is correct, the score is 1. If the second one is right, the score is 0.6309. The score decreases with the position of the right result moving backward. If none of the five predictions is right, then the score is 0.

III. Model selecting

A. baseline

We used the baseline from Airbnb. Since the NDF and US count over 80 percent of the training set, the baseline only predicts these two countries. It predicts the NDF and US alternatively, like NDF, US, NDF, US, ...etc. The prediction score on the validation set was 0.78640. The provided baseline seems somewhat trivial, but it got a good score compared to other baseline model we created by ourselves like linear regression, which just got a low score of 0.49859. Therefore, it was a valuable baseline to beat with our classification model.

B. feature representation and comparison

1) age: We first represented the age with its number directly, which gave us a trivial result. We then found that it was age distribution intervals that were more important, so we change our strategy into using vectors to represent the age. We discarded the training data with $age < 5$ and $age \geq 100$, then created a 20-dimension vector and initialized it to be $[0] * 20$, counted the index as $age/5$, and changed the value in this index to be 1 and remained others as 0. This gave us a good result.

2) date difference: date difference=date first booking−date account created.

We first used vector to represent date difference, which did less improvement to the result. There are very large positive and negative values. They count little but do have a negative effect on the result, so we discarded these outliers. The date difference < 11 count most of the date difference distribution, and it is the values of these date differences that have an evident different properties among all the countries. So we created a 12-dimension vector, used the index as date difference itself if it was ≤ 10 or used index 11 otherwise. This feature representation gave us a good prediction result.

3) first device type, first browser, signup app: We first treated these three features separately, which did no improvement on the model. After analysis, we found that they had relations when it comes to the loyalty to Apple products, such as safari, iPhone, iPad. So we used a 3-dimension vector to combine them all, which can reflect an encapsulation of a user's web browsing habit of loyalty to Apple. This feature representation gave us a good improvement.

4) affiliate provider, first affiliate tracked : Their separation gave us trivial results, so we combine them altogether as a 2-dimension vector, which could also reflect a user's habit.

C. classification method

For this classification task, we used linear regression, logistic regression, SVM from class, combining some other methods such as polynomial regression, two-level classification, two-class classifier for multiclass classification, voting mechanism and 1-r(one-against-rest) approach, probability logistic regression.

D. model selection

The data is unbalanced because US counts for a large proportion of the data. If we used multiclass classification classifiers directly, the performance was a disaster. Therefore, we built a two-level classifier to separate the US and other countries first.

For the first-level classifier, every single classification model performed not so well in the validation set, either nearly over-fitting or inaccurate, so we tried to combine them. The Voting Mechanism was then introduced to combine the three classification model. They increased the prediction accuracy drastically. The details of Voting Mechanism is on Section 4.

For the second-level classifier, it was a multi-class classification task. The evident method was to use SVM, but the normal SVM could only predict one result, while in this task we could predict at most 5 results. So we chose SVM's probability prediction to get the top 5 countries with the highest probabilities. We also considered using logistic regression for multi-class classification. We trained one classifier for every country to separate it from other countries, and used the classifier to predict this country's probability. Then we also selected the top 5 countries with the highest probabilities. Both second-level models performed well in the validation set. Next, we tried to combine them. This got a better result.

E. Description of the Model

As it is shown in figure 5, all the NDF have missing values for their "date-first-booking" feature,which indicates that we could use whether "date-first-booking" is equal to null to predict whether it is NDF.Therefore, we could discarded NDF prediction and focus on prediction of Non-NDF.

For the remaining training data which destination is Non-NDF, the count of US-users is 50699 while the count of Non-US-users is 21388. US accounts for a large proportion. Therefore we proposed a two-level classification model. In the first level, a binary classifier is built to distinguish US from Non-US, while a multi-class classifier was used to classify other countries in the second level. The model was shown as figure 10.

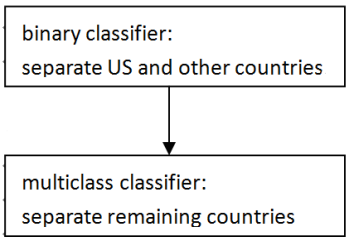


Fig. 10: Two-level classification model

For the first-level classification, we adopted Voting Mechanism taking into consideration linear regression, logistic regression and polynomial regression.(The reason why we don't use SVM is it's expensive.) We labeled US as 1 and other country-destination as 0.

For linear regression, if the prediction was equal to or larger than 0.5, then predicted US. For logistic regression, the result was either 1 or 0, if it was 1, then predicted US. Since data and labels may have non-linear relationship, we added polynomial regression with degree of 2.

If a classifier's prediction is US, then US gets one vote. And if US gets more than 1 vote out of 3 votes, then the output of Voting Mechanism is US, otherwise it's other. The strategy is shown as the Fig. 11.

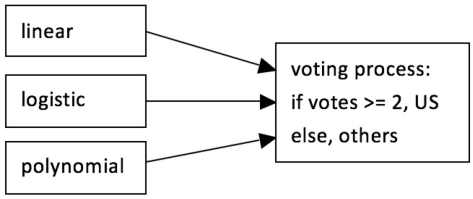


Fig. 11: first-level classifier: voting mechanism with linear, logistic and polynomial regression

For the second-level classification, we should separate the remaining 10 countries. We combined SVM and logistic regression. One important thing is that we could have at most 5 ordered predictions for each user. If one prediction was right, the wrong ones ordered after it had no effect on the prediction accuracy. Therefore, we used both SVM and logistic regression to make 5 predictions. For the SVM, we used it to predict the probabilities of all the countries directly, and select top 5 countries with highest probabilities. For the logistic regression, we used 1-r(one-against rest) approach. For each country, we trained a binary classifier to separate this country and all other countries. Then we could used a specified classifier to predict a country's probability. Then again, we chose the top 5 countries with the highest probabilities. Then, we got 10 (country, probabilities) pairs, normalized the 5 pairs from logistic regression. Finally, sorted 10 pairs in descending order and chose 5 unrepeated countries from the beginning.The model is shown as in the figure 12

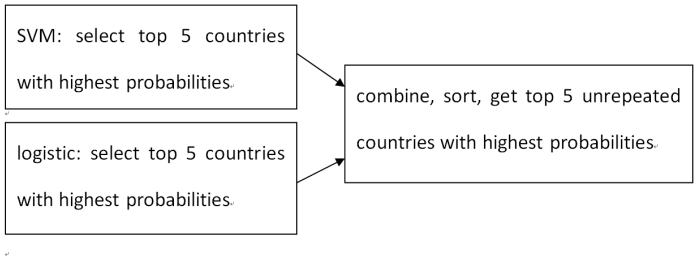


Fig. 12: second-level classifier: top 5 probability selection model with SVM and 1-r approach logistic regression

F. parameter tuning

Considering unbalanced data, we used a 'auto'('balanced') parameter to give different weight for each country, otherwise the model was weak. For the first-level classifier. We used greedy method, which was selecting the best-performed parameter for each single classifier - linear, logistic and polynomial regression, and then combing all three models. This got a good result. If we could tuned the parameters for the combination model directly, it would certainly give us a better result. However, it was very difficult and we still could not find a way to solve this problem. For the second-level classifier, we also used greedy method. We selected the good parameters for the SVM and multi-class logistic classifier alternatively and then combined their result. The result from SVM was the probabilities of each countries among all 10 countries , while the result from logistic was just the probability of one country to all other countries, so we normalized the logistic results, which gave us comparable probabilities from these two classifiers.

IV. Literature

The Airbnb dataset is the real-world data that comes from Airbnb company's kaggle competition. New users on Airbnb can book a place to stay in 34,000+ cities across 190+ countries. By accurately predicting where a new user will book their first travel experience, Airbnb can share more personalized content with their community, decrease the average time to first booking, and better forecast demand. So in this competition, Airbnb challenges people to predict in which country a new user will make his or her first booking.

Because the dataset comes from an ongoing competition, it has never been used on any other research before. However, there is a similar dataset that has been well-studied on the similar task. In [1], it proposed an accurate approach for another kaggle multi-classification prediction task—-Allstate Purchase Prediction Challenge(APPC), which is quite similar with our prediction task. As a customer shops an insurance policy, he/she will receive a number of quotes with different coverage options before purchasing a plan. In APPC, the task

model	nDCG
polynomial regression	0.92762
logistic regression	0.91660
linear regression	0.91552
Voting Mechanism	0.93230

TABLE II: Evaluation for different models

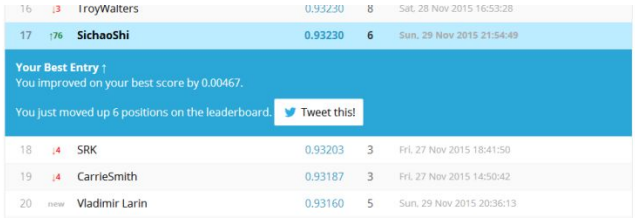


Fig. 13: ranking in kaggle

is to predict the purchased coverage options using a limited subset of the total interaction history.

In [1], it involves a super interesting method of prediction, called "Voting Mechanism"(VM). In VM, it proposed a problem that for some test data, it may have better prediction in a worse model, while having a worse prediction in a better model. To solve it, VM adopts several models. Some may perform good on validation data, some may not. And it lets models to vote based upon their own predictions. For example, [1] adopts Logistic Regression, SVM, Random forest, Last quoted policy(a model designed specifically for APPC problem). For a coverage option, if 3 models predict "yes", and 1 model predicts "no", then of course output is "yes". In my opinion, it may be better if the weight is assigned to each vote according to the performance of the each model.(Though the parameters are very difficult to be toned if there is too many models.)

The VM is very helpful when it's difficult to determine which model is better on all cases.(Maybe the performance of the model with the best accuracy is only a little better than the second one. Maybe for some extreme cases, their performance is bad on the best model, but commonly good on other models.) By the way, the author of [1] won the first prize of APPC based on VM.

As for the state-of-the-art methods of multi-classification, it's impossible to design one best model for all different datasets. A good model should be designed according to the actual properties and characters of the specific dataset. However, there is no doubt that SVM is a common method for multi-classification on most datasets. Besides that, Voting Mechanism provides a satisfying and comprehensive model by combining different models. In both [1] and our project, VM does the great job to prove itself.

V. Conclusion

Among three models, polynomial regression performs best. And as for linear and logistic regression, they perform a little worse. Nevertheless, when combining three models into the Voting Mechanism, it outperforms all three models.

The result of all models are shown in Table.II. And the ranking in kaggle is shown as Fig.13. The ranking is 17 out of 177, and the difference of nDCG among the top rankings is less than 0.001.(Which I think with different test cases, the ranking could be totally different.)

As for the feature representation, the details are shown in Section.2. And the vector feature representation of 0,1 performs better than digit feature representation on most features. For example, when represent "age" feature, the way to split the age into intervals of size 5 and then represent it with a vector of 0,1, where 1 means the age is in this interval, outperforms the way to use the number of age as the feature representation directly.

In VM, there is several different parameters for different member models, therefore it's improper to give the comprehensive interpretation for a set of parameters. But the interpretation of each single model is available. In linear regression, the parameter is the penalty for weight in regularization to avoid overfitting and underfitting problem. In polynomial regression, the parameter is the degree of polynomial features. In logistic regression, the parameter class_weight is 'auto', which would automatically adjust weights inversely proportional to class frequencies in the input data. As for SVM, the parameter C represents the penalty of the error term.

According to Table.II, VM outperforms other methods. And the reason for it is quite obvious that VM overcomes the flaws that appear in the single model. For example, linear regression doesn't perform well when the relation between features and result is not linear, but this defect doesn't matter in logistic and polynomial model. As for logistic and polynomial regression, they may generate disappointing result when processing some extreme cases, which could be precessed well on linear regression.

For the further work, we may assign the weight to each model in VM in order to improve the accuracy. Though the assignment2 is done, but the kaggle competition is still ongoing.

References

[1] Saba Arslan Shah, Mehreen Saeed, Predicting Purchased Policy for Customers in Allstate Purchase Prediction Challenge on Kaggle

[2] D. Tax, R. Duin, Using two-class classifiers for multi-class classification, in: International Conference on Pattern Recognition, Quebec City, QC, Canada, August 2002.

[3] Chong Wang, Y. W. (2012). Discovering Consumers Behavior Changes Based on Purchase Sequences. 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012) (pp. 642 - 645). IEEE.

[4] Jiawei Han, H. C. (2007). Frequent pattern mining: current status and future directions. Data Mining and Knowledge Discovery, pp 55-86

[5] Pedregosa, F. a. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2825-2830.

[6] R. A. Fisher. The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7(2):179-188, 1936.

[7] T. Hastie and W. Stuetzle. Principal curves. Journal of the American Statistical Association, 84(406):502-516, June 1989.

[8] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research, 2:263-286, 1995.