INFO7390 Final Project: Hawaii Tourism Prediction

Group 8:

Zhao Wang: Lezi Wang: Qiaoyi He

Topic- Hawaii Tourism Prediction

- As we all know, tourism industry is the largest capital source of Hawaii economy.
- The end user of our project is Hawaii government and related workers
- **MODEL1** - Goal: Predict monthly visitor amount in each island from multiple countries, diversify Hawaii's global and domestic major markets.
- **MODEL2** - Goal: Predict monthly total visitor amount for one specific island in Hawaii area, in order to build more accurate prediction models for different islands.
- **MODEL3** - Goal: Predict monthly total visitor amount in entire Hawaii area, enhance strategic plans to incorporate marketing programs that drive travel demand, visitor arrivals and spending.
- **MODEL4** - Goal: Predict monthly total visitors' expenditures in entire Hawaii area, in order to enhance and promote the profits of Hawaii's tourism industry.

Data Set Source

- 1.Get Hawaii monthly visitor records from Hawaii government website.
- http://dbedt.hawaii.gov/visitor/tourism/
- 2. Get Hawaii temperature records from US climate websites.
- http://www.usclimatedata.com/climate/honolulu/hawaii/united-states/ushi0026
- 3. Get US monthly vacation days from timeanddate.com.
- http://www.timeanddate.com/holidays/us/
- 4. Get Hawaii monthly tourism incomes (total visitors' expenditures) data from Hawaii Tourism website.
- http://www.hawaiitourismauthority.org/research/reports/historical-visitor-statistics/

Pre-Process data

- We use R to pre-process the data
- 1. Uniform units (the unit of expenditure is million and the unit of visitors' amount is ten thousand)
- 2.Separation time into year and month
- 3. Add More variable (average maximum temperature, average minimum temperature and average temperature, and vacation day of each month, island and country)Clean data
- 4. Clean data
- 5.Combine data set

Model 1 Data Set

The data set contains two additional columns – island and country for predict monthly visitor number in each island from multiple countries,

	A	B	C	D	E	F	G	H	I
1	Year	Month	Visitors	Average hig	Average low	Average ten	extra vacatic	Island	Country
2	2007	1	115535.638	80.9	68.8	74.85	2	HawaiiIslanc	Canada
3	2007	2	100557.7	80.2	66.7	73.45	1	HawaiiIslanc	Canada
4	2007	3	95819.4767	80.5	67.9	74.2	0	HawaiiIslanc	Canada
5	2007	4	52189.8731	83.6	69.7	76.65	1	HawaiiIslanc	Canada
6	2007	5	35086.8204	85	71.6	78.3	0	HawaiiIslanc	Canada
7	2007	6	26549.1139	87.3	74.1	80.7	1	HawaiiIslanc	Canada
8	2007	7	32061.7345	88	75.2	81.6	0	HawaiiIslanc	Canada
9	2007	8	45759.3657	88.3	75.8	82.05	0	HawaiiIslanc	Canada
10	2007	9	33578.4448	88.2	74.9	81.55	1	HawaiiIslanc	Canada
11	2007	10	49379.9716	86.3	74	80.15	1	HawaiiIslanc	Canada
12	2007	11	71148.8788	82.7	70.5	76.6	2	HawaiiIslanc	Canada
13	2007	12	99489.1121	80	71.1	75.55	2	HawaiiIslanc	Canada

Model 2 Data Set

The data set contains an additional columns – island for predict monthly total visitor number for one specific island in Hawaii area.

	A	B	C	D	E	F	G	H
1	Year	Month	total_vistors	Average hig	Average low	Average ten	extra vacatic	Island
2	2007	1	195264.608	80.9	68.8	74.85	2	Maui
3	2007	2	196700.12	80.2	66.7	73.45	1	Maui
4	2007	3	227232.515	80.5	67.9	74.2	0	Maui
5	2007	4	202215.773	83.6	69.7	76.65	1	Maui
6	2007	5	198130.154	85	71.6	78.3	0	Maui
7	2007	6	241790.41	87.3	74.1	80.7	1	Maui
8	2007	7	247535.244	88	75.2	81.6	0	Maui
9	2007	8	237113.276	88.3	75.8	82.05	0	Maui
10	2007	9	186111.351	88.2	74.9	81.55	1	Maui
11	2007	10	190684.617	86.3	74	80.15	1	Maui
12	2007	11	184472.898	82.7	70.5	76.6	2	Maui
13	2007	12	214791.747	80	71.1	75.55	2	Maui

MODEL 3 DATA SET

Predict monthly total visitor number in entire Hawaii area.

	A	B	C	D	E	F	G	H
1	Year	Month	Total_Visitor	expenditure	Average hig	Average low	Average ten	extra vacatior
2	2007	1	577231.793	1089.87397	80.9	68.8	74.85	2
3	2007	2	574762.708	996.76546	80.2	66.7	73.45	1
4	2007	3	674532.008	1028.06454	80.5	67.9	74.2	0
5	2007	4	597477.56	957.542315	83.6	69.7	76.65	1
6	2007	5	586545.552	922.25895	85	71.6	78.3	0
7	2007	6	672585.524	1135.55192	87.3	74.1	80.7	1
8	2007	7	711263.325	1191.92992	88	75.2	81.6	0
9	2007	8	733025.281	1177.58518	88.3	75.8	82.05	0
10	2007	9	558430.761	911.175938	88.2	74.9	81.55	1
11	2007	10	570646.621	969.321885	86.3	74	80.15	1
12	2007	11	576370.975	950.496904	82.7	70.5	76.6	2

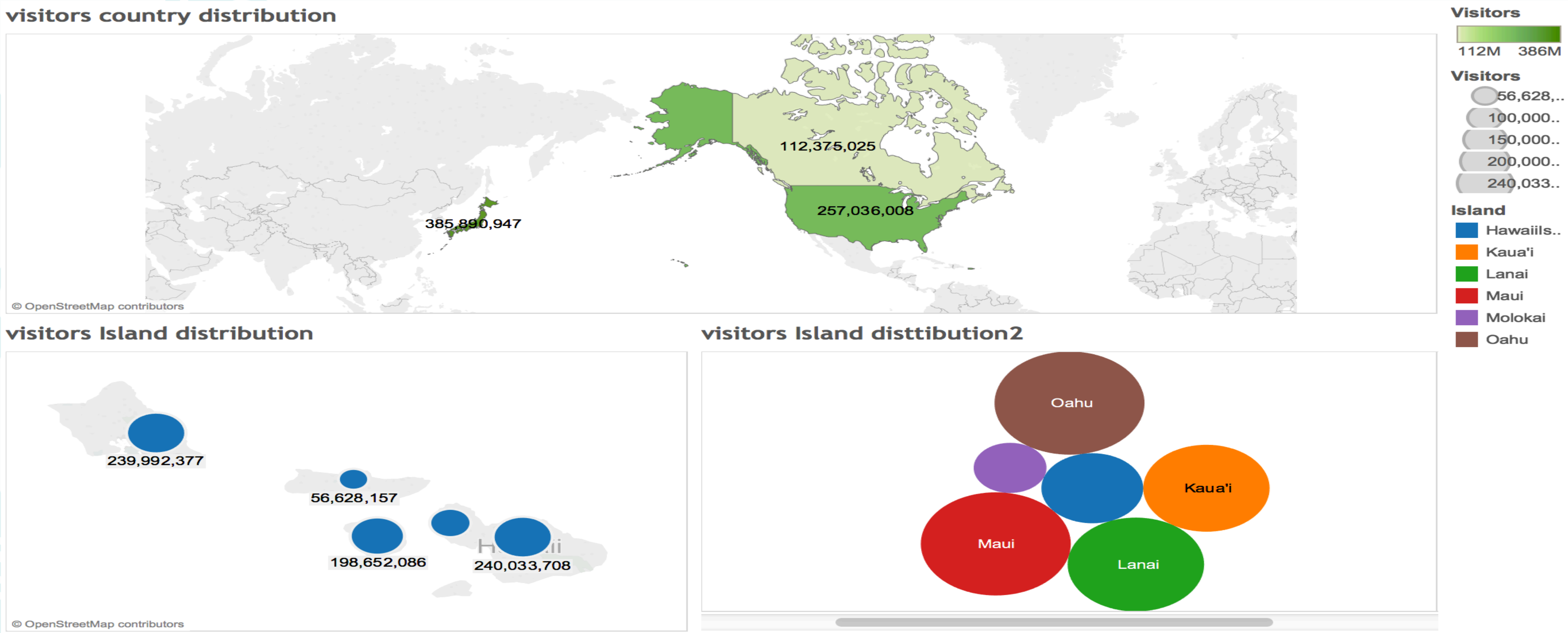
MODEL 4 DATA SET

Predict monthly total visitors' expenditures in entire Hawaii area.

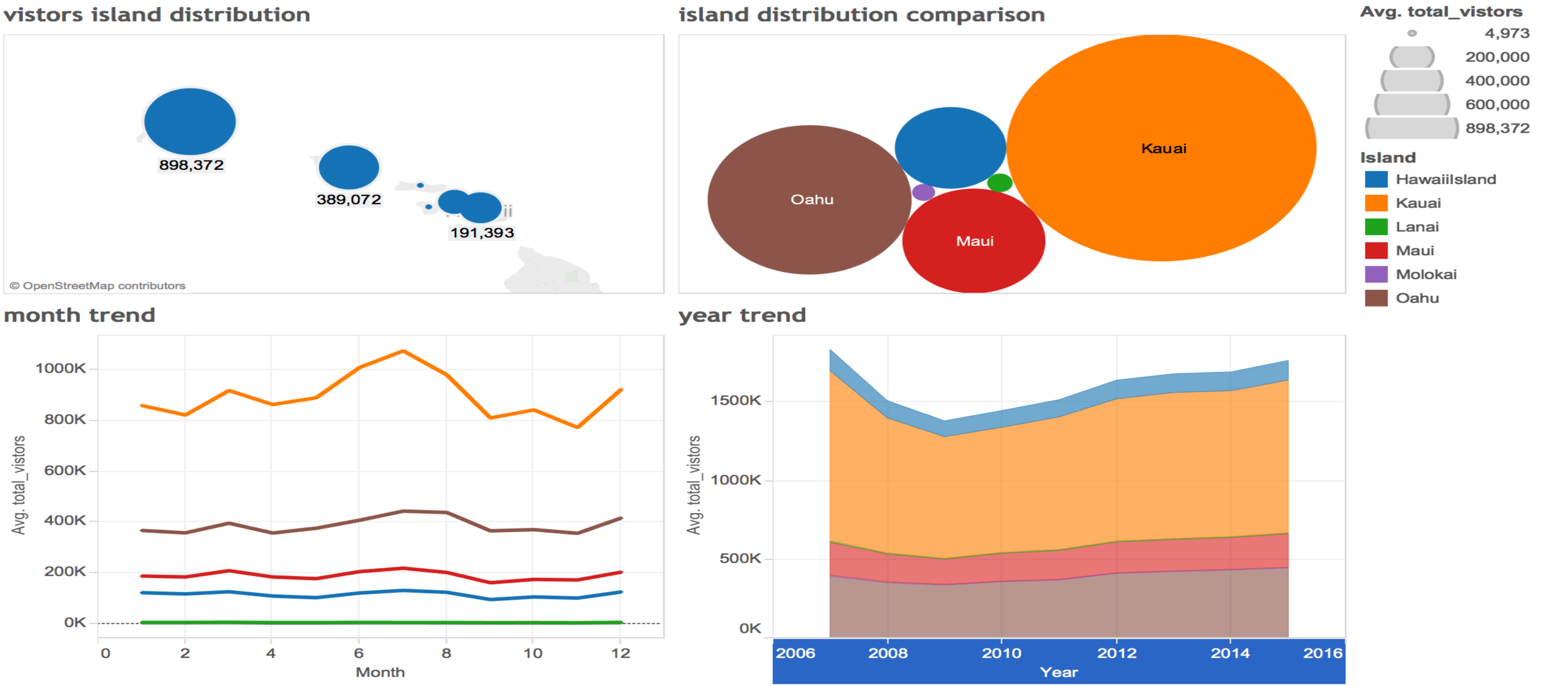
	A	B	C	D	E	F	G	H
1	Year	Month	Total_Visitor	expenditure	Average hig	Average low	Average ten	extra vacatio
2	2007	1	577231.793	1089.87397	80.9	68.8	74.85	2
3	2007	2	574762.708	996.76546	80.2	66.7	73.45	1
4	2007	3	674532.008	1028.06454	80.5	67.9	74.2	0
5	2007	4	597477.56	957.542315	83.6	69.7	76.65	1
6	2007	5	586545.552	922.25895	85	71.6	78.3	0
7	2007	6	672585.524	1135.55192	87.3	74.1	80.7	1
8	2007	7	711263.325	1191.92992	88	75.2	81.6	0
9	2007	8	733025.281	1177.58518	88.3	75.8	82.05	0
10	2007	9	558430.761	911.175938	88.2	74.9	81.55	1
11	2007	10	570646.621	969.321885	86.3	74	80.15	1
12	2007	11	576370.975	950.496904	82.7	70.5	76.6	2

Visualization-Model1

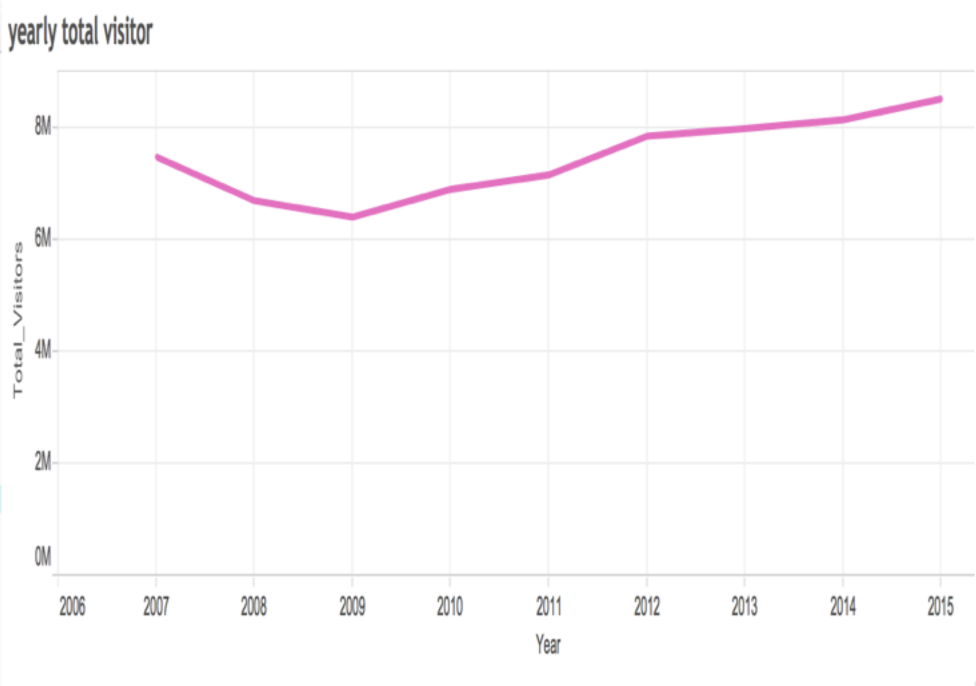
– We do visualization for each model separately.

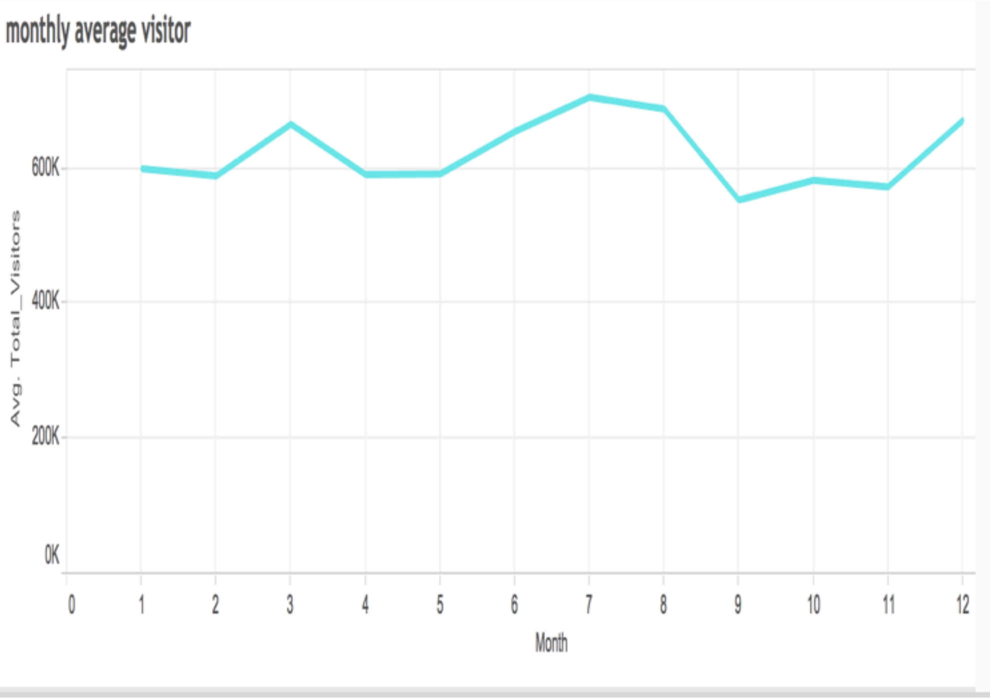


Visualization-Model2



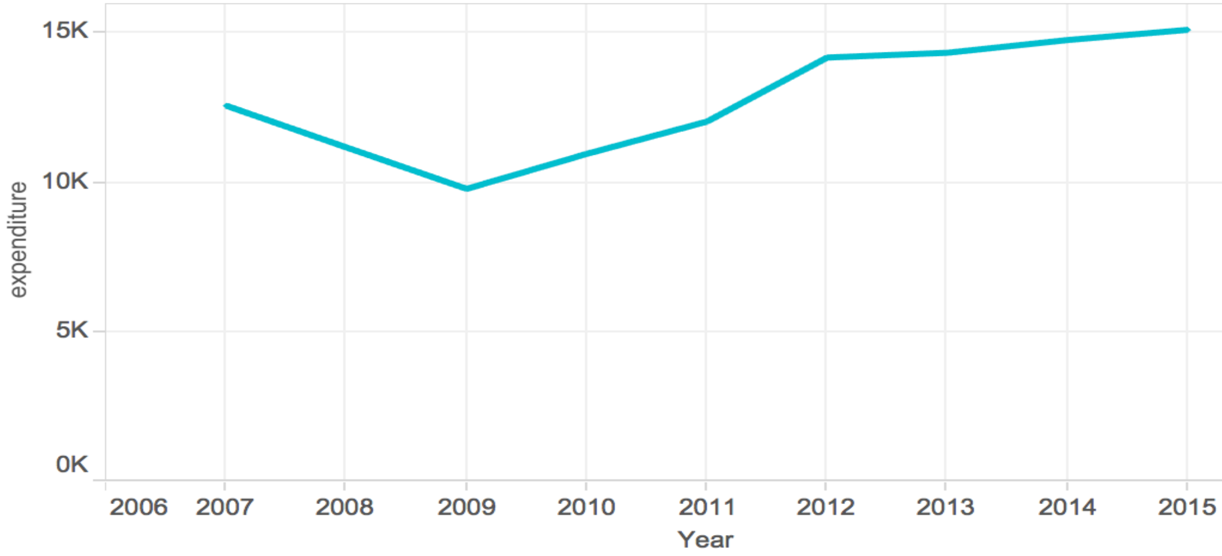
Visualization-Model3



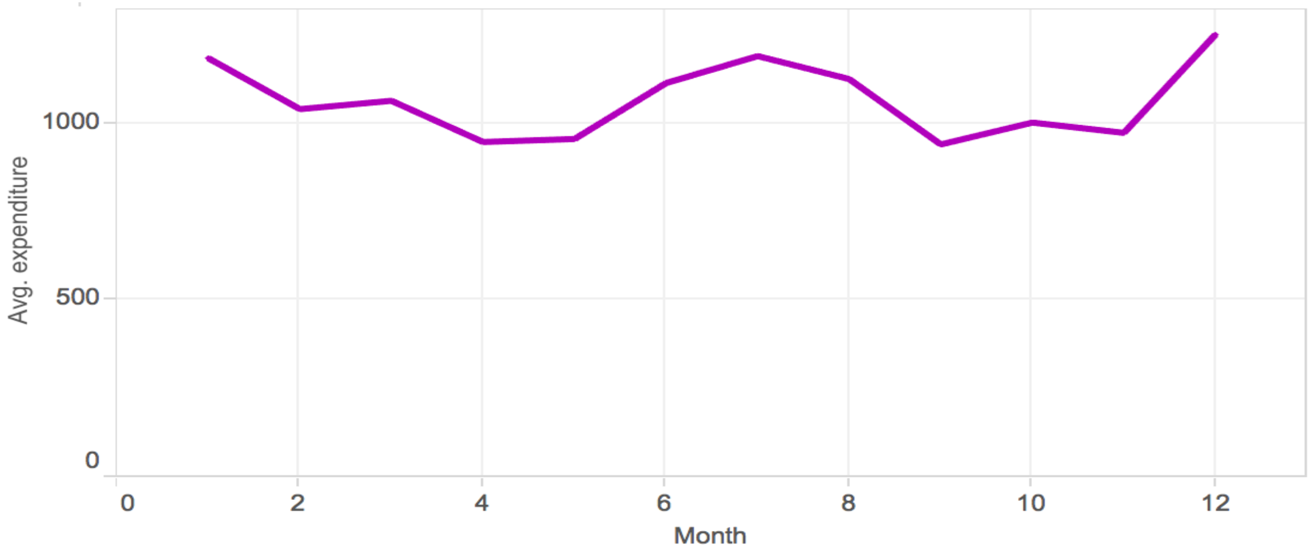


Visualization-Model4

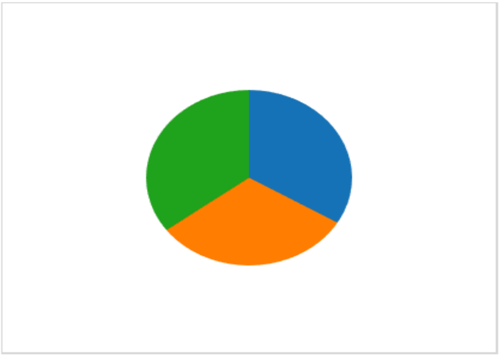
yearly expenditure



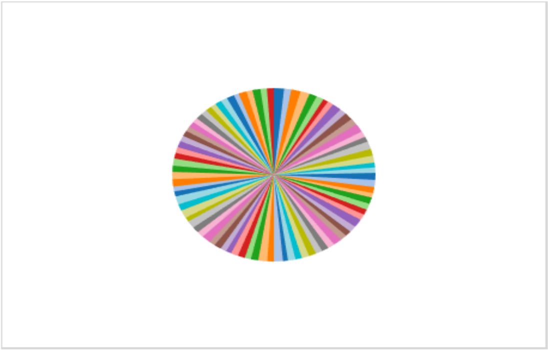
Monthly expenditure



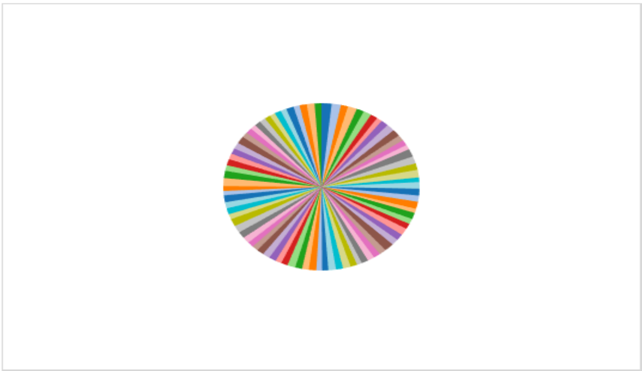
vacation & expenditure



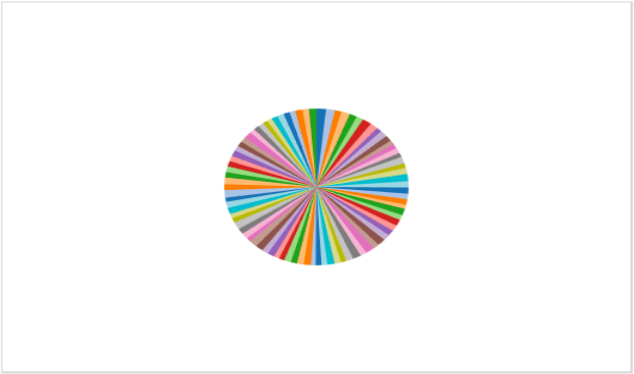
high temperature & expenditure



average temperature & expenditure



low temperature & expenditure

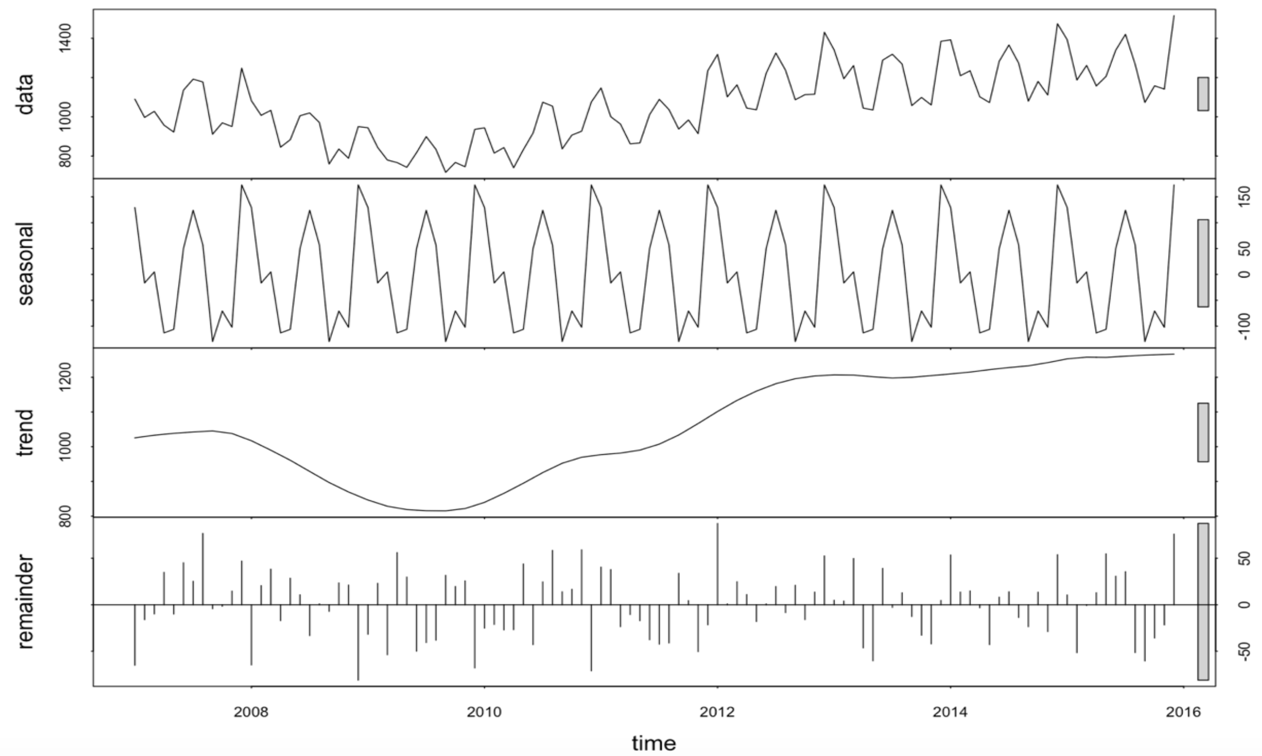


Model4: Time series - R

Predict monthly total visitors' expenditures in entire Hawaii area.

```
data1<-read.csv("~/Desktop/total_hawaii_expenditure.csv")
#transform data to time series, unit is month, start time point is jan 2007
tdata<- ts(data1[[3]],start=c(2007,1), frequency = 12)
```

```
#Seasonal Decomposition of Time Series by Loess
plot(stl(tdata,s.window="periodic"))
```



Partition Data

- We separate the data set into training dataset and testing dataset according to time. We use the data from 2007 to 2014 for building model, and use 2015 data to validate the result. The following picture is the R code which we use to separate the data set.

```r
#2007-2014 as traindata, 2015 as valitation data
traindata<-window(tdata,start=2007,end=2014+11/12)
testdata<-window(tdata,start=2015)
```

Exponential smoothing state space model(ETS)

– We use four different function - Ses, holt, hw and ets, to build the Exponential smoothing state space model. In fact, Ses, holt and hw are simply convenient wrapper functions for forecast(ets(...)). And for the fit1, since we not specify the model, R returns the best model automatically. The following picture is the R code and the note of accuracy is the RMS of the model.

```
pred_holt<-holt(traindata,h=12,damped=F,initial="simple",beta=0.65)
accuracy(pred_holt)#166
plot(pred_holt)

pred_ses <- ses(traindata,h=12,initial='simple',alpha=0.2)
accuracy(pred_ses)#123
plot(pred_ses)

pred_hw<-hw(traindata,h=12,seasonal='multiplicative')
accuracy(pred_hw)#42.8
plot(pred_hw)

fit1<-ets(traindata)
accuracy(predict(fit1,12),testdata) #42.43
plot(fit1)
```

Arima

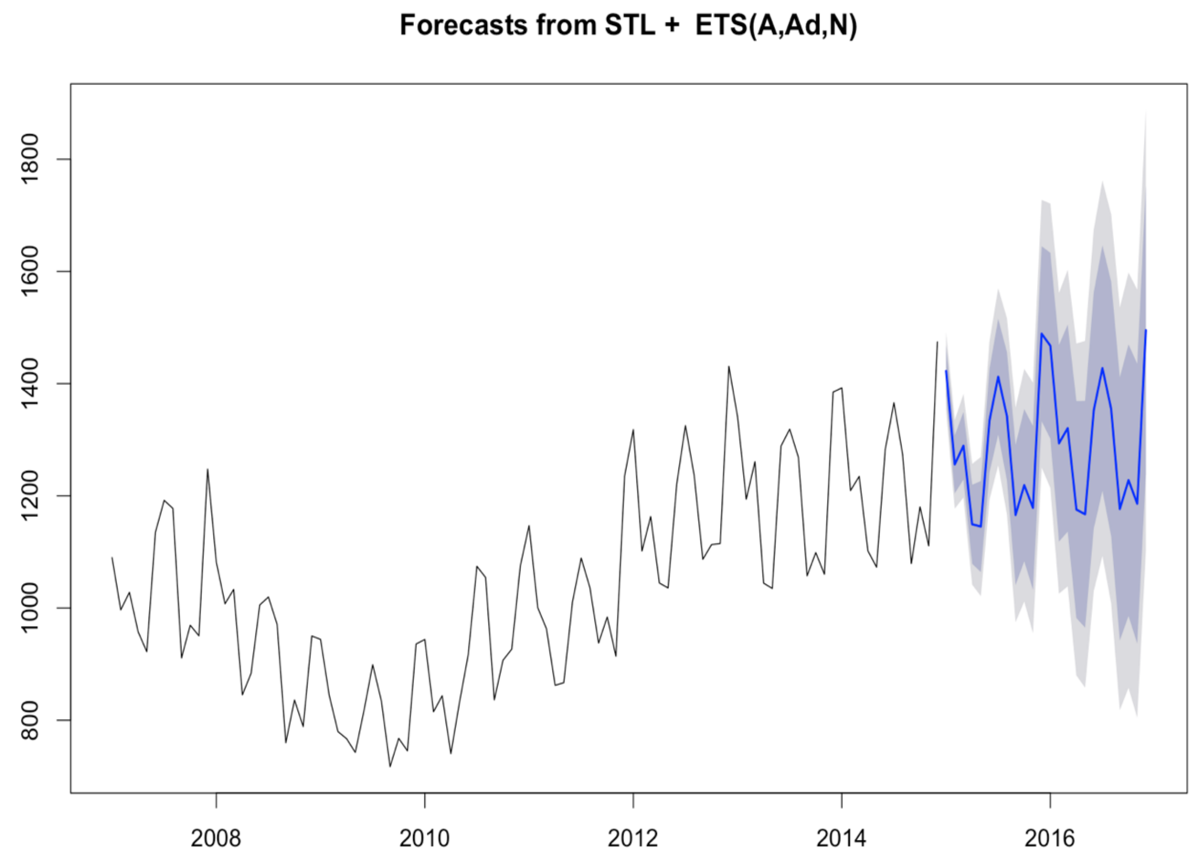
– We use four different function-naive, snaive, arima and auto.arima , to build the ARIMA model. Naive() returns forecasts and prediction intervals for an ARIMA(0,1,0) random walk model applied to x. Snaive() returns forecasts and prediction intervals from an ARIMA(0,0,0)(0,1,0)m model where m is the seasonal period. The different between the auto.arima function and the rest is the auto.arima returns best ARIMA model according to either AIC, AICc or BIC value. So the only argument auto,arima need is dataset. The following picture is the R code and the note beside accuracy is the RMS of the model.

```r
pred_naive<-naive(traindata,h=12)
accuracy(pred_naive)#132

pred_snaive<-snaive(traindata,h=12)
accuracy(pred_snaive)#123
plot(pred_snaive)

fit2<-auto.arima(traindata)
accuracy(forecast(fit2,h=12),testdata) #45
plot(fit2)

ma = arima(traindata, order = c(0, 1, 3),   seasonal=list(order=c(0,1,3), period=12))
p<-predict(ma,12)
accuracy(p$pred,testdata)  #48
```

STLF

– Stlf combines STL decomposition and ETS model.

```
#stl+ets(AAN)
pred_stlf<-stlf(traindata)
accuracy(pred_stlf)#34.99
plot(pred_stlf)
```



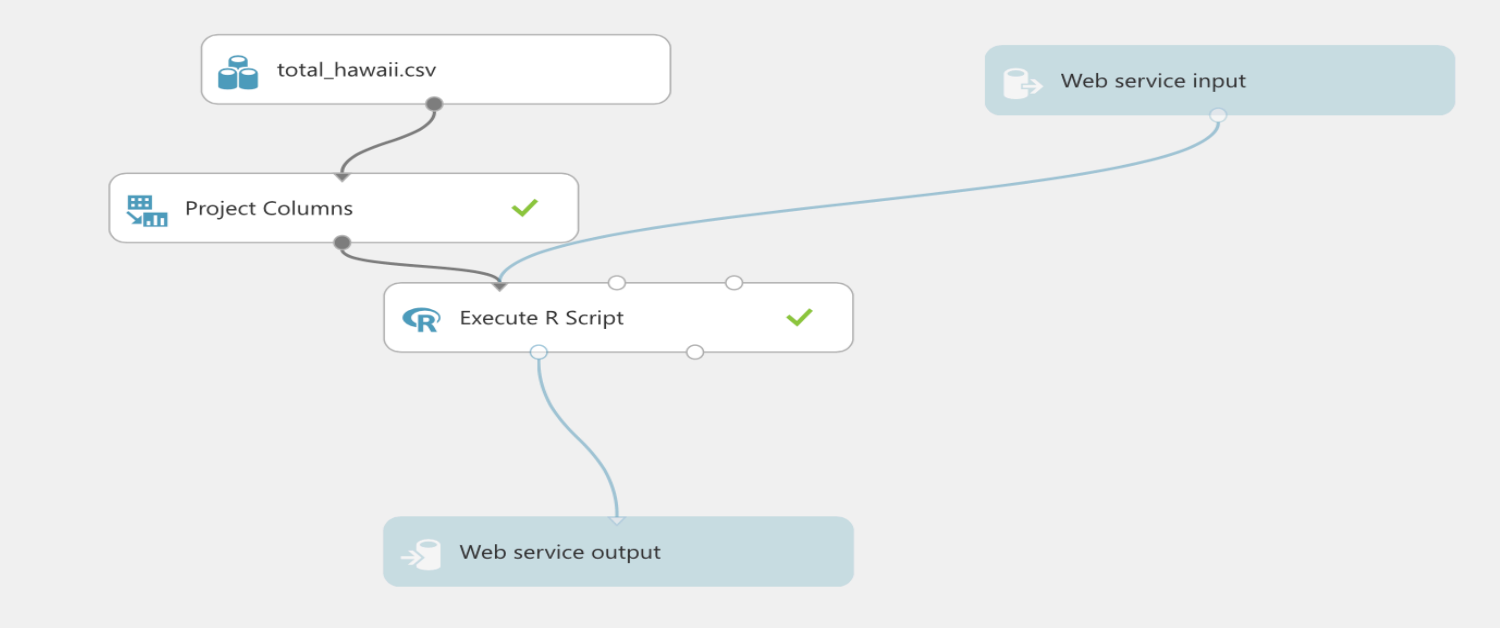
The deep gray area represents the 80% forecast period, and the light gray area represents the 95% forecast period.

Compare models

```
> accuracy(predict(fit1,12),testdata) #42.43
                    ME     RMSE      MAE         MPE     MAPE      MASE
Training set  1.224696 42.43927 32.16605  0.08735142 3.139882 0.3239598
Test set    -48.025355 68.14964 57.95313 -3.92560033 4.756453 0.5836740
                    ACF1 Theil's U
Training set 0.004679037        NA
Test set     0.624958536 0.4194878
> accuracy(forecast(fit2,h=12),testdata) #45
                    ME     RMSE      MAE        MPE     MAPE      MASE
Training set  2.510318 45.80851 34.83629  0.1975113 3.498749 0.3508531
Test set    -52.219994 67.69040 59.87994 -4.3156069 4.950865 0.6030799
                     ACF1 Theil's U
Training set -0.009153377        NA
Test set      0.358379796 0.4143118
> accuracy(pred_stlf)#34.99
                   ME     RMSE      MAE       MPE     MAPE      MASE
Training set 2.014956 34.99689 28.13921 0.1217105 2.798023 0.2834036
                    ACF1
Training set 0.002465656
```

- fit1 - ETS function,
- fit2 - Arima function,
- pred_stlf - STLF function.
- As you can see, the third model has lower ME, RMSE, MAE, MAPE and MASE than ARIMA model. What’s more, compare to ETS functions, STLF function can avoid seasonality being ignored. The ets models is a better choose if the data are non-seasonal or the seasonal period is 12 or less and if the seasonal period is 13 or more stlf is a better option.

Building model in azure



Year	Month	Total_Visitors	expenditure	Average high temperature	Average low temperature	Average temperature	extra vacation
2007	1	577231.7925	1089.873966	80.9	68.8	74.85	2
2007	2	574762.7083	996.76546	80.2	66.7	73.45	1
2007	3	674532.0081	1028.064539	80.5	67.9	74.2	0
2007	4	597477.5602	957.542314	83.6	69.7	76.65	1

R code

```
# Map 1-based optional input ports to variables
data1 <- maml.mapInputPort(1) # class: data.frame
library(forecast)
#transform data to time series
tdata<- ts(data1[[3]],start=c(2007,1), frequency = 12)
#use all data to building model
pred_stlf<-stlf(tdata)
#predict the mean and 95% forecast period.
Forecast <- pred_stlf$mean
Lo95 <- pred_stlf$lower[,1]
Hi95 <- pred_stlf$upper[,1]
# Select data.frame to be sent to the output Dataset port
output<-data.frame(cbind(Forecast, Hi95, Lo95),Month=1:12,Year=20
maml.mapOutputPort("output");
```

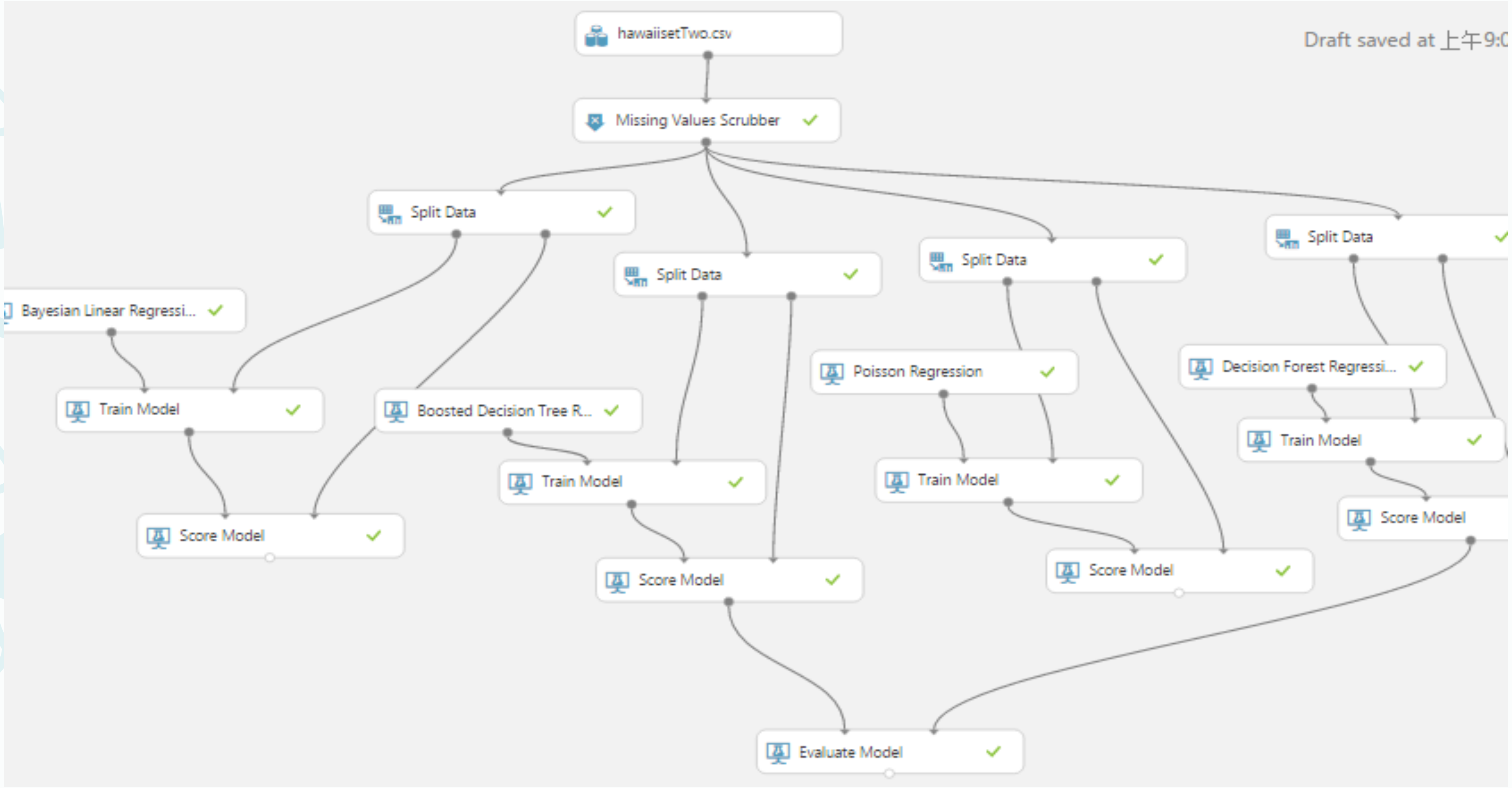
Output

rows 24 columns 5

view as

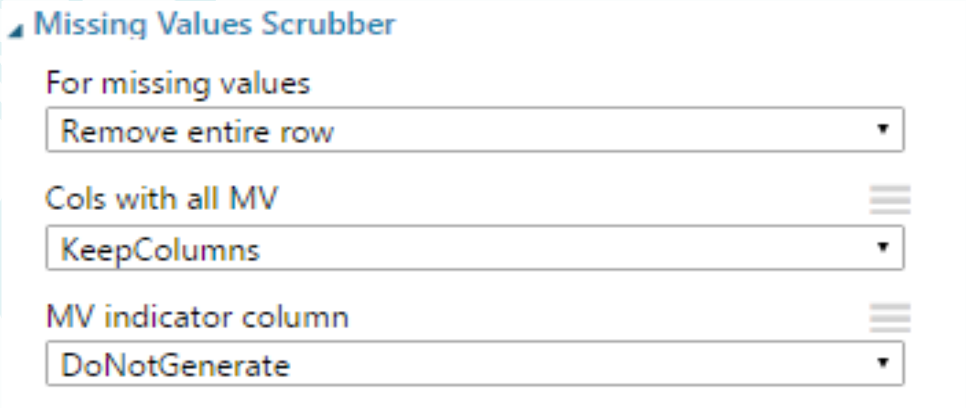
Forecast	Hi95	Lo95	Month	Year
1429.025321	1474.506843	1383.543798	1	2016
1248.16645	1300.850846	1195.482054	2	2016
1296.184481	1357.993297	1234.375664	3	2016
1156.996789	1229.07144	1084.922137	4	2016
1159.435679	1242.36237	1076.508989	5	2016
1346.297722	1440.306006	1252.289438	6	2016
1419.317028	1524.414627	1314.21943	7	2016
1324.675684	1440.734174	1208.617193	8	2016
1139.452167	1266.261048	1012.643287	9	2016
1202.3756	1339.676736	1065.074464	10	2016
1164.448152	1311.957992	1016.938311	11	2016
1503.50419	1660.928214	1346.080167	12	2016
1456.85083	1623.892951	1289.808709	1	2017
1271.238436	1447.607085	1094.869787	2	2017
1315.315003	1500.726972	1129.903033	3	2017
1172.85918	1367.04197	978.676391	4	2017
1172.588246	1375.281373	969.895119	5	2017
1357.203392	1568.15901	1146.247775	6	2017
1428.359647	1647.342686	1209.376608	7	2017
1332.173522	1558.961522	1105.385522	8	2017
1145.669125	1380.051853	911.286397	9	2017
1207.530495	1449.30944	965.751549	10	2017
1168.722419	1417.710212	919.734625	11	2017
1507.04827	1763.068063	1251.028476	12	2017

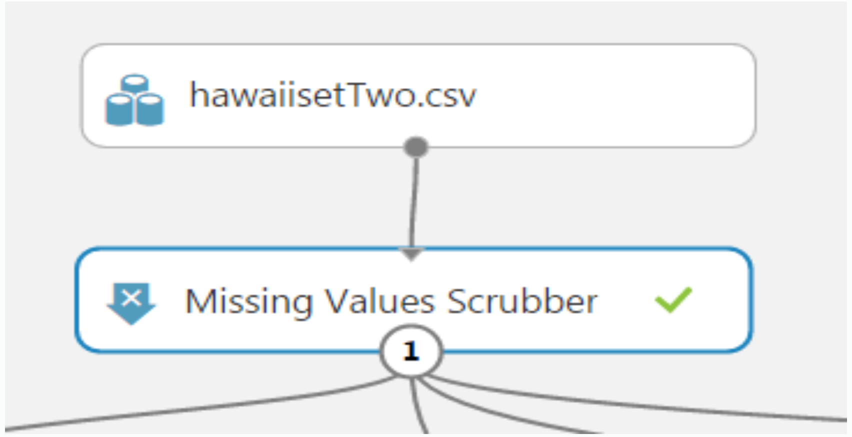
Comparing Algorithms



Get dataset and Pre-process

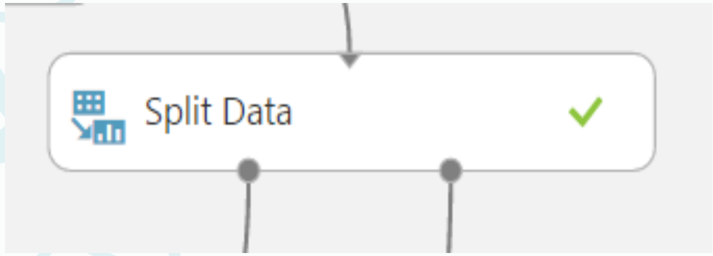
- We used the dataset which is saved in Azure Machine Learning Studio to build prediction models.
- After importing the dataset, we used the Missing Values Scrubber to remove the entire row which contains missing data. Because missing data will affect the prediction result, we must deal with it before building models. Sometimes, you may replace missing data with median or average of that column, but for this model we remove that row.

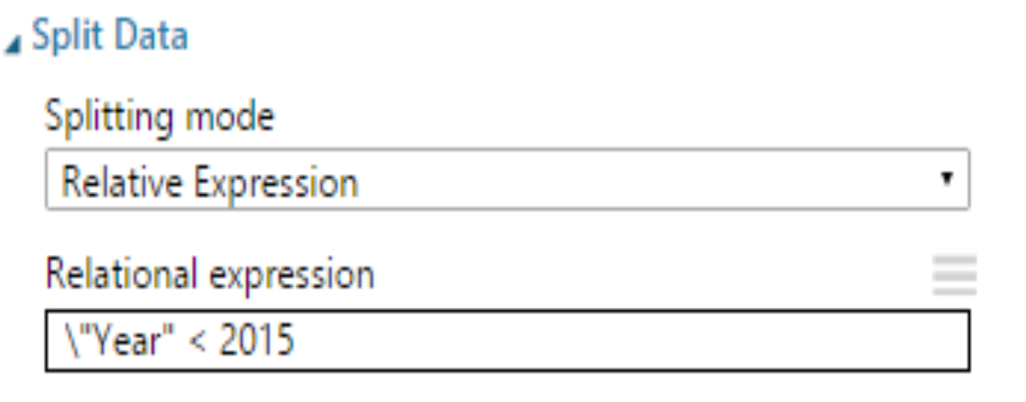




Split Data Based on Year

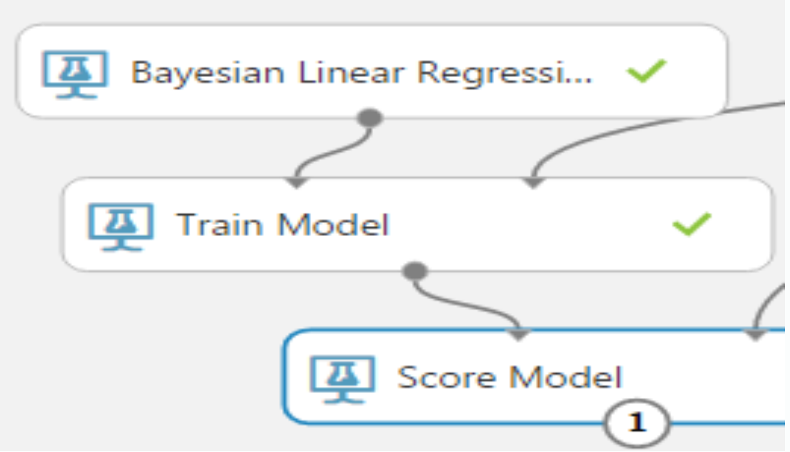
– Because these data satisfy the requirements of Time Series Model, we followed the rule of Time Series Model for splitting data—splitting data based on Year. In our dataset, there are 9 years records (2007-2015). Thus, we split these data into 2007-2014's records for training data and 2015's recodes for validation data. As you can see in right picture, the Relation expression is "Year <2015", this is for splitting 2007-2014's records from all records.





Train Model with Bayesian Linear Regression

– The first algorithm is Bayesian Linear Regression. In statistics, it is an approach to linear regression in which the statistical analysis is undertaken within the context of Bayesian inference. When the regression model has errors that have a normal distribution, and if a particular form of prior distribution is assumed, explicit results are available for the posterior probability distributions of the model's parameters.



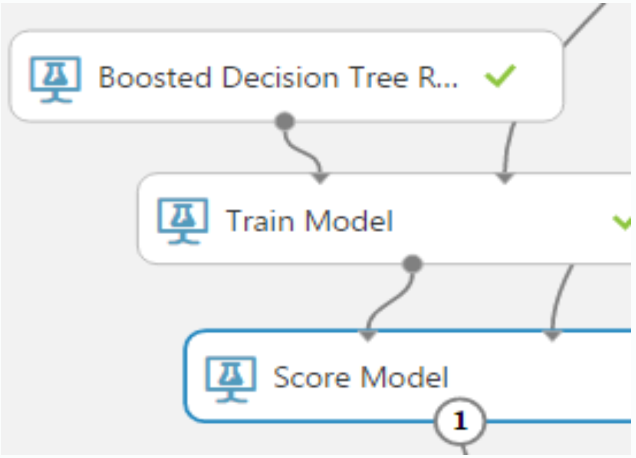
Bayesian Linear Regression Score

– The Score of this algorithm model posted above, you can the difference between the real value and the mean of predicted value is not small. The differences are almost 50000, and the unit is myriad(Ten Thousand). For scoring part, this algorithm does not have a good performance.

Year	Month	total_vistors	Average high temperature	Average low temperature	Average temperature	extra vacation	Island	Scored Label Mean	Scored Label Standard Deviation
2015	1	214819.9659	62.9	48.2	55.55	2	Maui	162386.747054	100049.021087
2015	2	198999.3332	68.8	53.9	61.35	1	Maui	182150.765689	99619.54011
2015	3	234905.7754	68	52.5	60.25	0	Maui	195160.433049	99743.255487
2015	4	205070.8646	72.2	55	63.6	1	Maui	183249.254028	99605.268362
2015	5	209508.309	65.6	51.5	58.55	0	Maui	195413.574995	100135.139456
2015	6	233046.9678	73.1	55.5	64.3	1	Maui	184360.600935	99600.431636
2015	7	245896.4667	75.5	60.3	67.9	0	Maui	204068.453916	99579.930467

Train Model with Boosted Decision Tree Regression

- The second algorithm is Boosted Decision Tree Regression. It enables to create an ensemble of regression trees using boosting. Boosting means that each tree is dependent on prior trees, and learns by fitting the residual of the trees that preceded it. Thus, boosting in a decision tree ensemble tends to improve accuracy with some small risk of less coverage. This regression method is a supervised learning method.



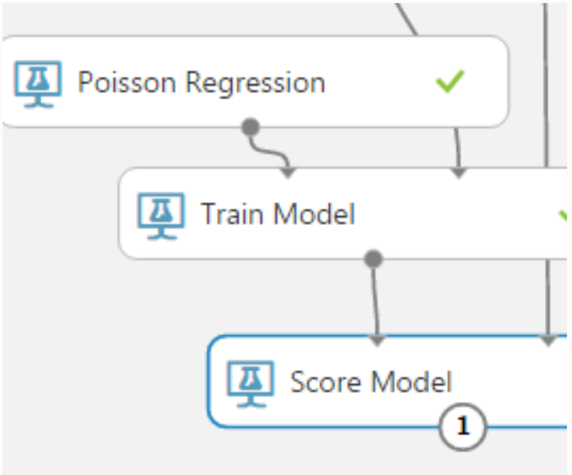
Boosted Decision Tree Regression Score

– The Score of this algorithm model posted above, you can the difference between the real value and the predicted value is not that much big. The differences are roughly 20000, and the unit is million, it smaller than the first algorithm’s. For scoring part, this algorithm has a normal performance.

Year	Month	total_vistors	Average high temperature	Average low temperature	Average temperature	extra vacation	Island	Scored Labels
2015	1	214819.9659	62.9	48.2	55.55	2	Maui	196203.671875
2015	2	198999.3332	68.8	53.9	61.35	1	Maui	201752.734375
2015	3	234905.7754	68	52.5	60.25	0	Maui	216471.078125
2015	4	205070.8646	72.2	55	63.6	1	Maui	192462.0625
2015	5	209508.309	65.6	51.5	58.55	0	Maui	192117.578125
2015	6	233046.9678	73.1	55.5	64.3	1	Maui	214409.453125

Train Model with Poisson Regression

– The third algorithm is Poisson Regression. In statistics, it is a form of regression analysis used to model count data and contingency tables. Poisson regression assumes the response variable Y has a Poisson distribution, and assumes the logarithm of its expected value can be modeled by a linear combination of unknown parameters. A Poisson regression model is sometimes known as alog-linear model, especially when used to model contingency tables.



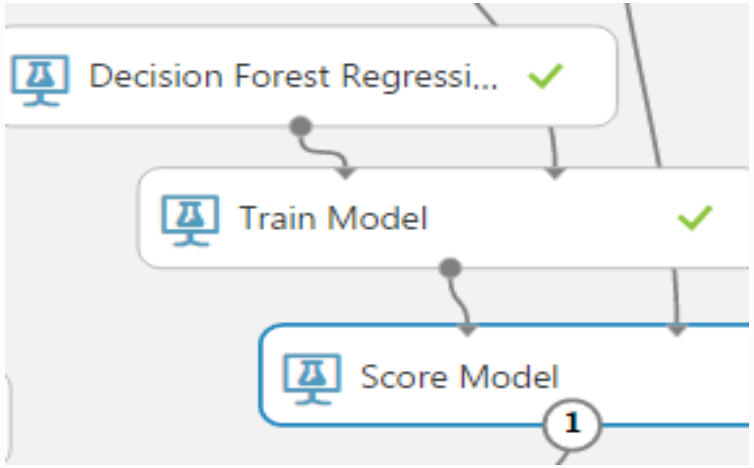
Poisson Regression Score

– The Score of this algorithm model posted above, you can the difference between the real value and the predicted value is not that much big. The differences are roughly greater than 30000, and the unit is million, it smaller than the first algorithm’s but bigger than second’s. For scoring part, this algorithm has a not that bad performance. So far, the second algorithm has the best performance in scoring part.

Year	Month	total_vistors	Average high temperature	Average low temperature	Average temperature	extra vacation	Island	Scored Labels
2015	1	214819.9659	62.9	48.2	55.55	2	Maui	166043.622711
2015	2	198999.3332	68.8	53.9	61.35	1	Maui	170234.342101
2015	3	234905.7754	68	52.5	60.25	0	Maui	174468.382847
2015	4	205070.8646	72.2	55	63.6	1	Maui	170243.712007
2015	5	209508.309	65.6	51.5	58.55	0	Maui	174459.649169
2015	6	233046.9678	73.1	55.5	64.3	1	Maui	170248.011367

Train Model with Decision Forest Regression

– The fourth algorithm is Decision Forest Regression. It is used to create a regression model using an ensemble of decision trees. Decision trees are non-parametric models that perform a sequence of simple tests for each instance, traversing a binary tree data structure until a leaf node (decision) is reached.



Decision Forest Regression Score

– The Score of this algorithm model posted above, you can the difference between the real value and the mean of predicted value is not that much big. The differences are roughly 20000, and the unit is million, it is similar with second algorithm. For scoring part, this algorithm has a normal performance similar with second algorithm.

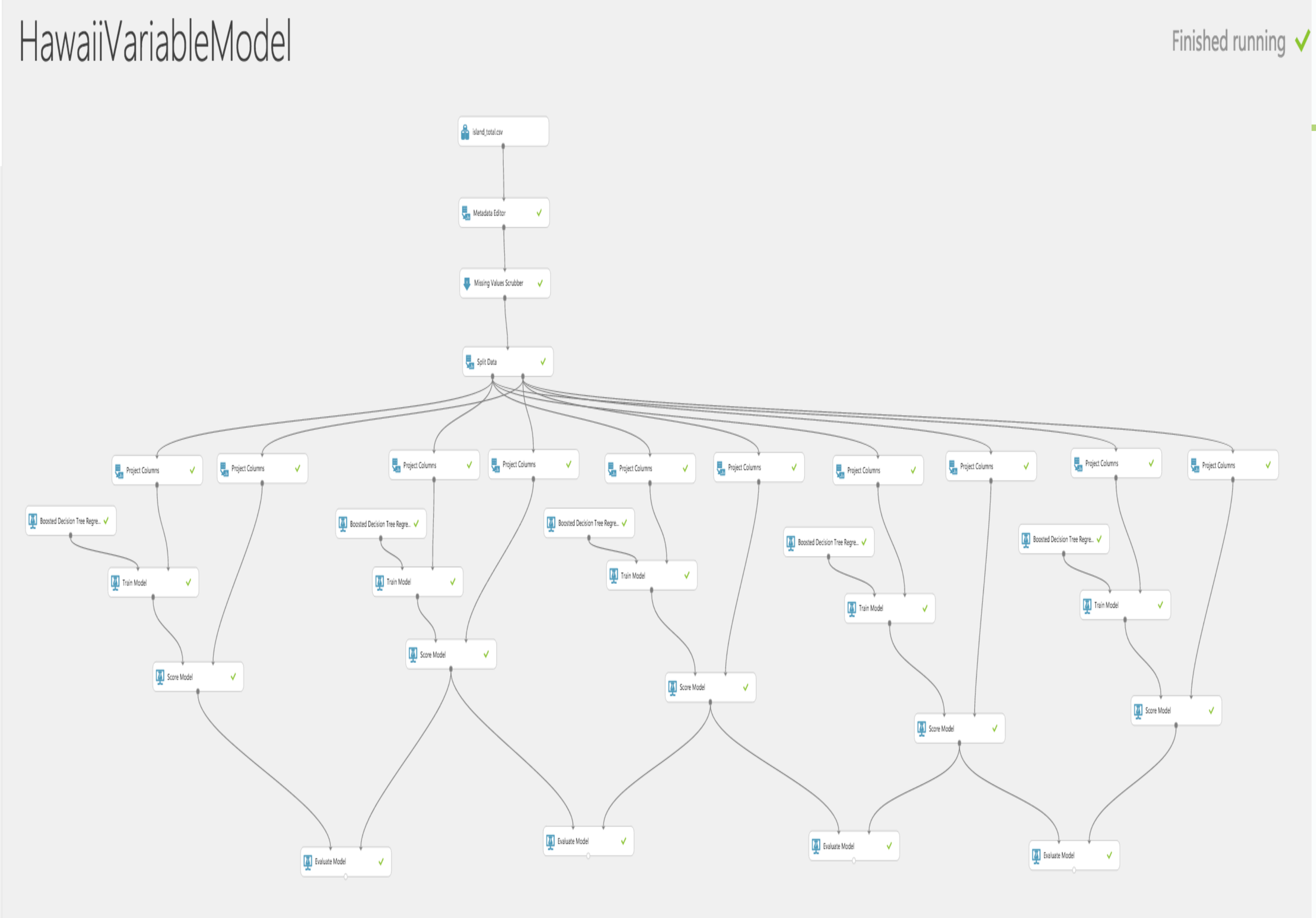
Year	Month	total_vistors	Average high temperature	Average low temperature	Average temperature	extra vacation	Island	Scored Label Mean	Scored Label Standard Deviation
2015	1	214819.9659	62.9	48.2	55.55	2	Maui	193362.504663	6618.615888
2015	2	198999.3332	68.8	53.9	61.35	1	Maui	195639.658442	5854.769274
2015	3	234905.7754	68	52.5	60.25	0	Maui	214312.728277	16912.30164
2015	4	205070.8646	72.2	55	63.6	1	Maui	188456.257519	14009.340933
2015	5	209508.309	65.6	51.5	58.55	0	Maui	209307.254381	20483.745813
2015	6	233046.9678	73.1	55.5	64.3	1	Maui	206927.208723	11699.270409

Comparing Algorithm Result

	Negative Log Likelihood	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
Model1	909.851096	40545.591246	64267.14665	0.147051	0.035679	0.964321
Model2	Infinity	17420.976855	30573.238088	0.063182	0.009166	0.990834

	Negative Log Likelihood	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
Model3	Infinity	54790.387504	77334.407763	0.198714	0.051663	0.948337
Model4	914.027199	16723.019723	31575.335785	0.060651	0.008613	0.991387

Variable Selection



– We build prediction models with different variables in Azure ML. We choose dataset2 (monthly total visitors in different islands) to build four kinds of monthly total visitor prediction models with different variable subsets, and then compare models' performance and choose the best variable subsets for further prediction models.

Variable Selection-Model1

– Use all 8 variables to build the prediction model, and the model performance is shown below. The RMS of this prediction model is 32573.

SELECTED COLUMNS

All Types | search columns

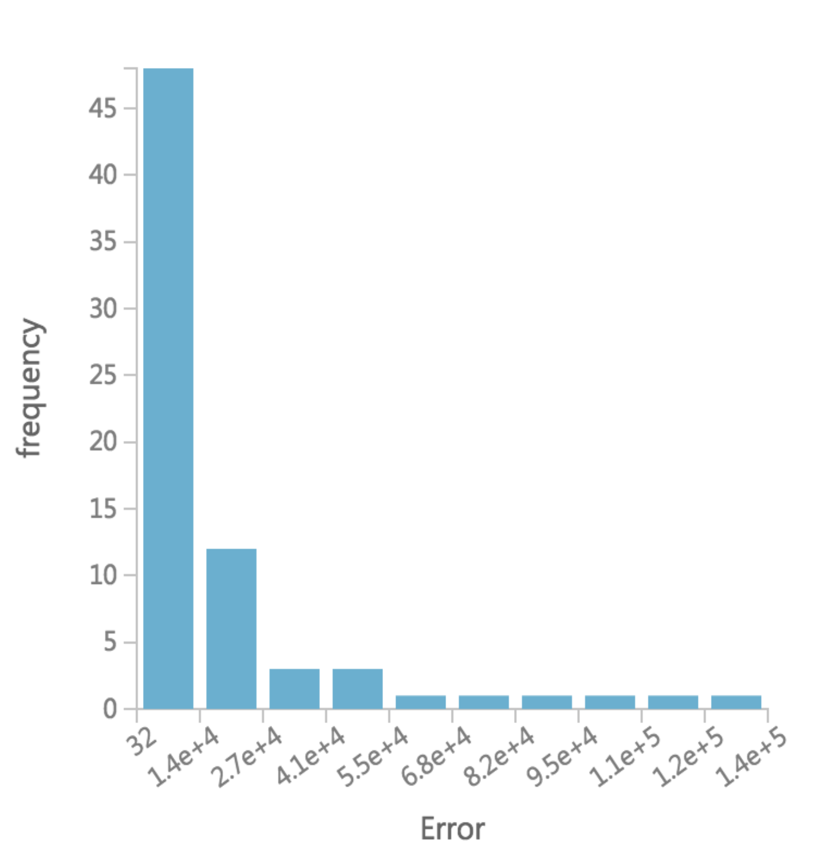
Year
Month
total_vistors
Average high temperature
Average low temperature
Average temperature
extra vacation
Location

8 columns selected

Metrics

Mean Absolute Error	17420.976855
Root Mean Squared Error	32573.238088
Relative Absolute Error	0.063182
Relative Squared Error	0.009166
Coefficient of Determination	0.990834

Error Histogram



Variable Selection-Model2

– Remove the average temperature variable to build the prediction model, and the model performance is shown below. The RMS of this prediction model is 29240 .

SELECTED COLUMNS

All Types | search columns

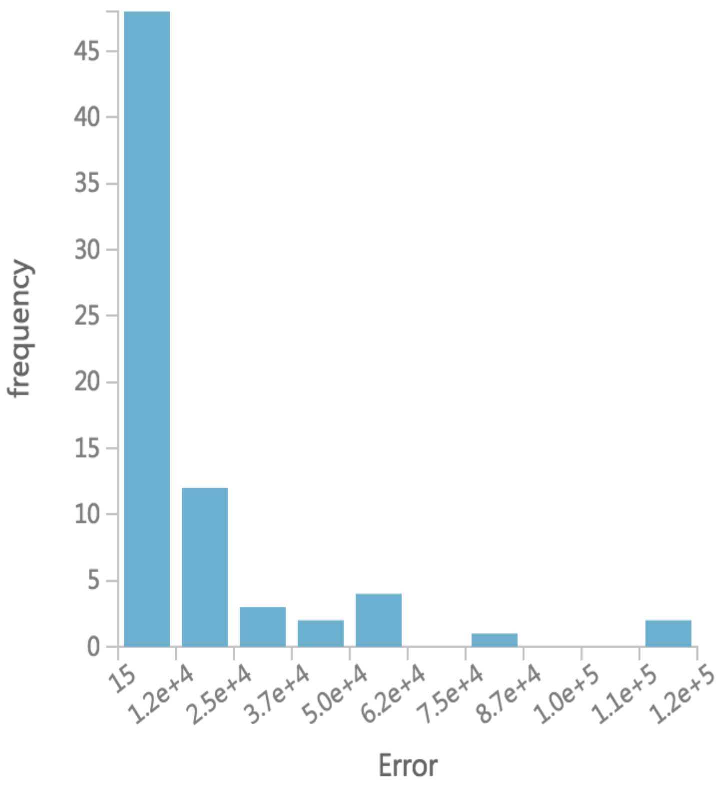
Year
Month
total_vistors
Average high temperature
Average low temperature
Location
extra vacation

7 columns selected

Metrics

Mean Absolute Error	16010.228788
Root Mean Squared Error	29240.003872
Relative Absolute Error	0.058066
Relative Squared Error	0.007386
Coefficient of Determination	0.992614

Error Histogram



Variable Selection-Model3

– Remove the high and low temperature variables to build the prediction model, and the model performance is shown below. The RMS of this prediction model is 30797 .

SELECTED COLUMNS

All Types | search columns

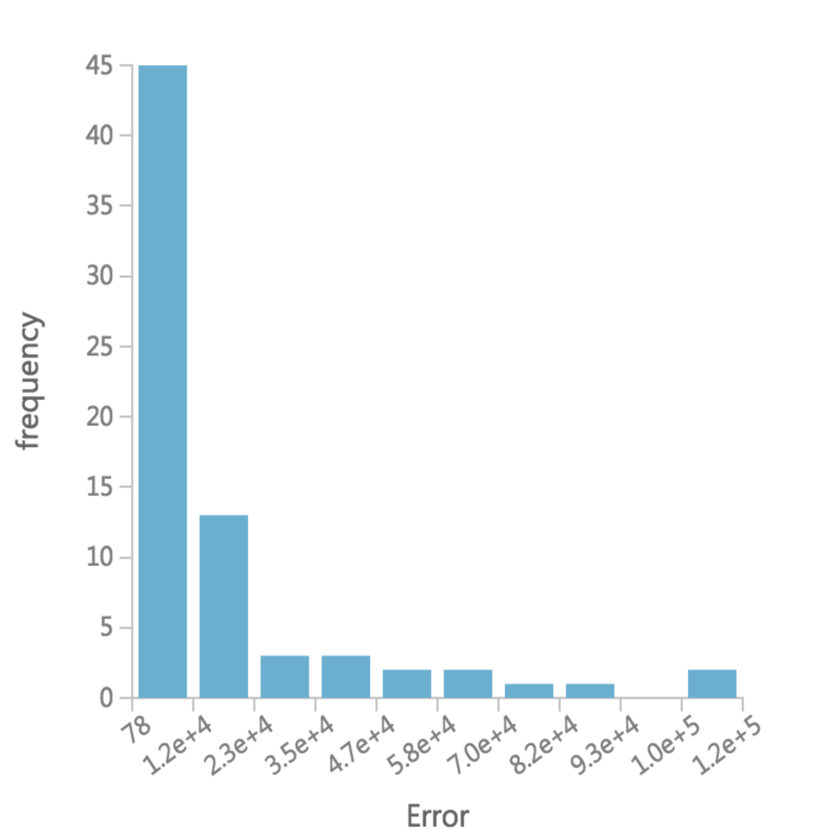
Year
Month
total_vistors
Location
extra vacation
Average temperature

6 columns selected

Metrics

Mean Absolute Error	17745.582965
Root Mean Squared Error	30797.980991
Relative Absolute Error	0.06436
Relative Squared Error	0.008194
Coefficient of Determination	0.991806

Error Histogram



Variable Selection-Model4

- We remove the extra vacation days variable to build the prediction model, and the model performance is shown below. The RMS of this prediction model is 28795 .

SELECTED COLUMNS

All Types | search columns

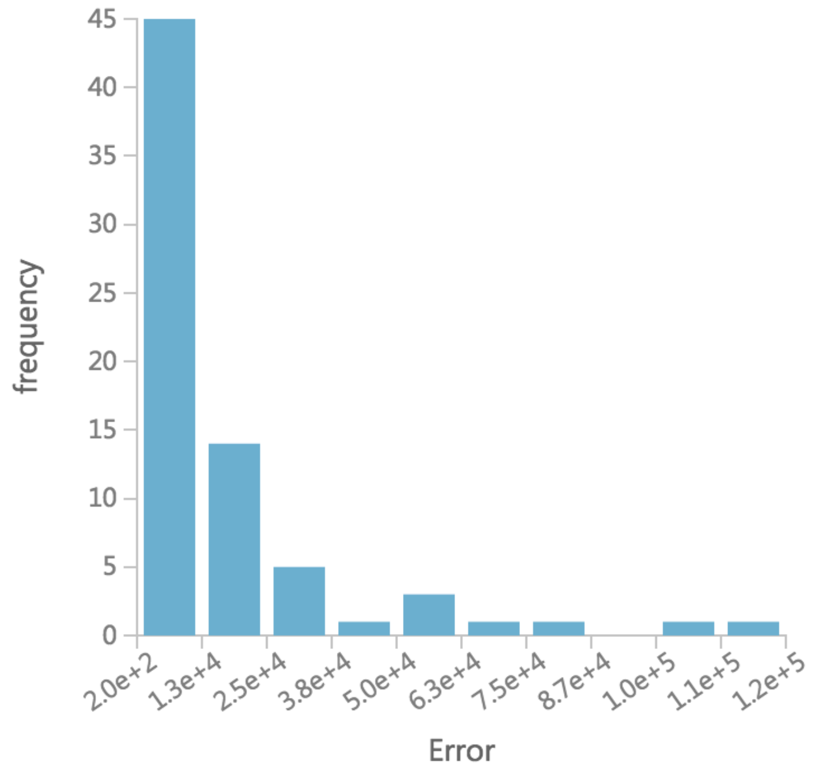
Year
Month
total_vistors
Average high temperature
Average low temperature
Average temperature
Location

7 columns selected

Metrics

Mean Absolute Error	16051.741462
Root Mean Squared Error	28795.895825
Relative Absolute Error	0.058216
Relative Squared Error	0.007163
Coefficient of Determination	0.992837

Error Histogram



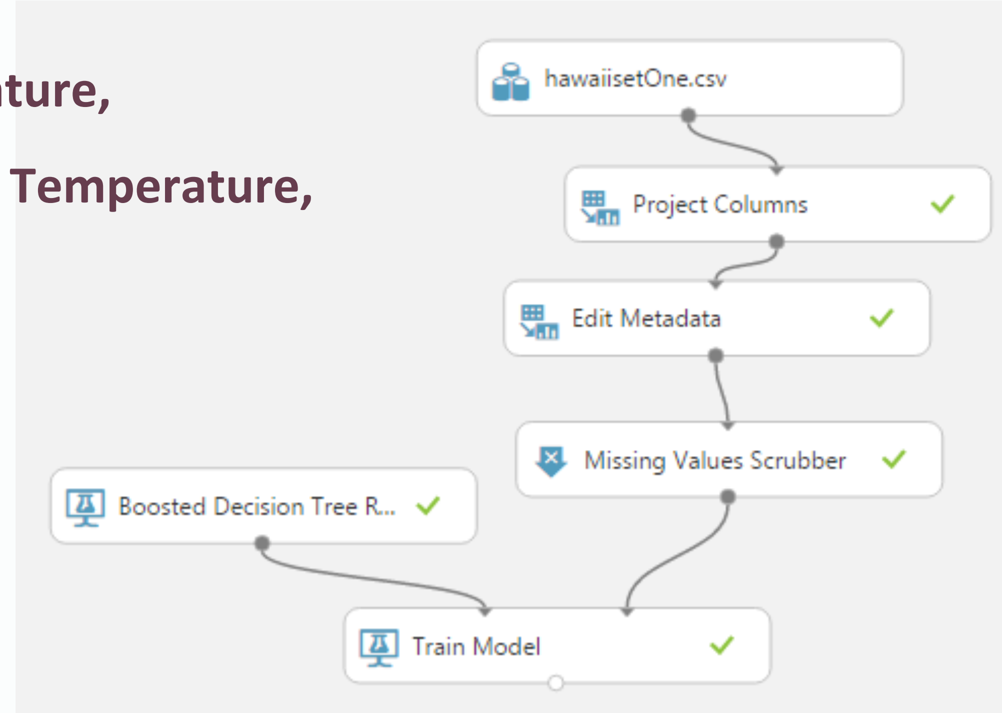
Variable Selection-Result

- After comparison, we decide to remove extra vacation day variable.
- choose following variables to build prediction models: Year, Month, monthly high temperature, monthly low temperature, monthly average temperature, Location (island name).

	Model1	Model2	Model3	Model4
RMS	32573	29240	30797	28795

Azure Machine Learning Model 1

- The first model is for predicting the Visitor Amount of Each Island from multiple countries.
- Input: Year, Month, High Temperature,
  Low Temperature, Average Temperature,
  Island, Country.
- Output : Visitor

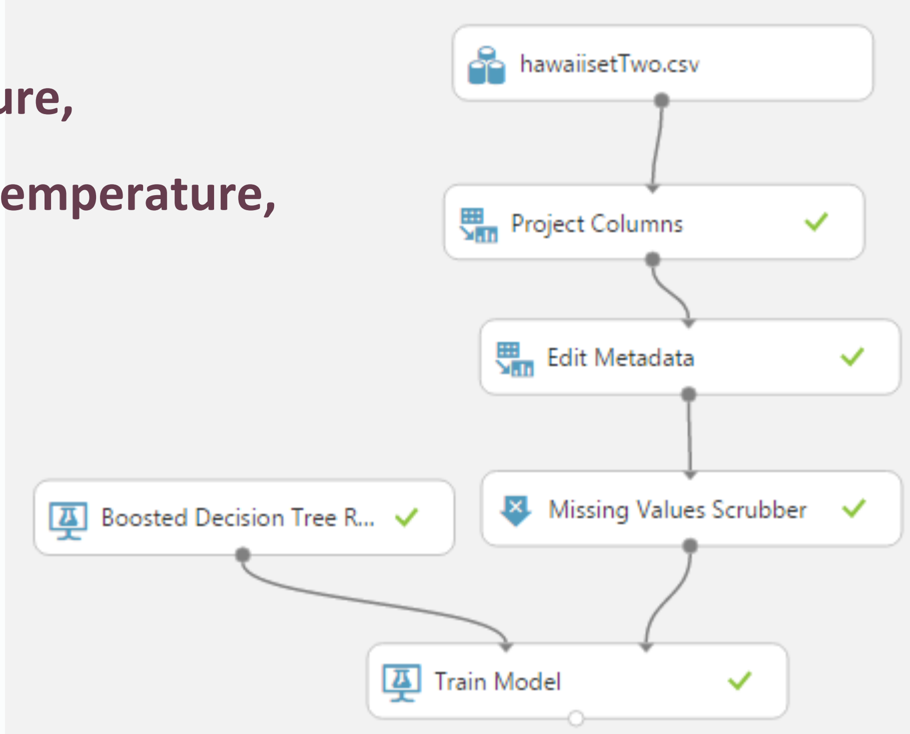


Model 2

- The second model is for predicting the Total Visitor Amount of Each Major Island of Hawaii
- Input: Year, Month, High Temperature,

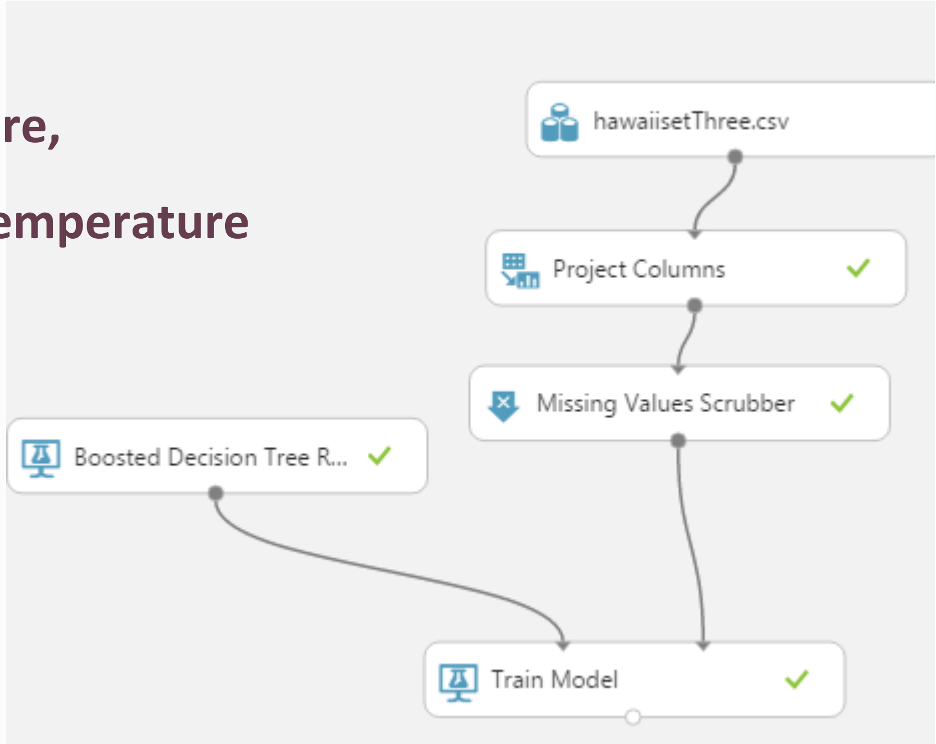
  Low Temperature, Average Temperature,

  Island.
- Output: Total Visitor



Model 3

- The third model is for predicting Total Visitor Amount of Hawaii for one future Month.
- Input: Year, Month, High Temperature,
  Low Temperature, Average Temperature
- Output: Total Visitor

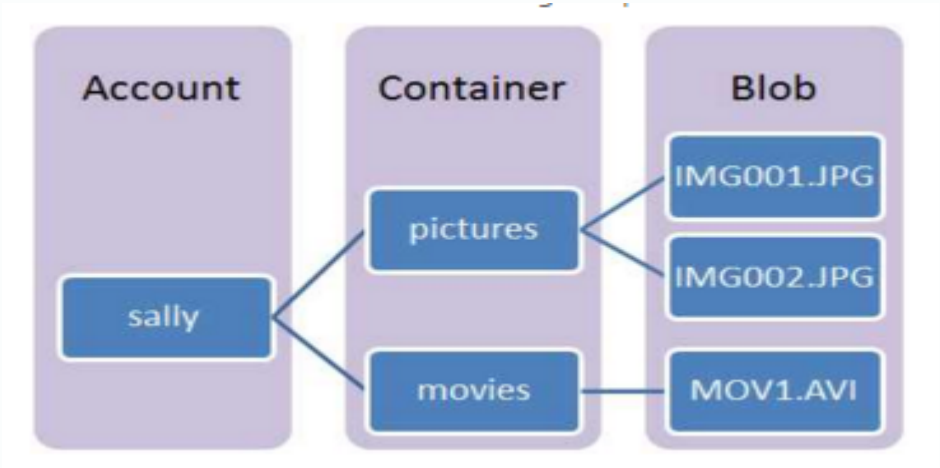


Deploy Web Service & Configuration
Deploy Web Service

- After predictive models were created, we deployed Web Service in Azure to get models' APIs and URIs. These two values are very important for developing an integration of model.
- API stands for application programming interface. It can be helpful to think of the API as a way for different apps to talk to one another. For many users, the main interaction with the API will be through API keys, which allow other apps to access your account without you giving out your password.
- To paraphrase the World Wide Web Consortium, Internet space is inhabited by many points of content. A URI (Uniform Resource Identifier; pronounced YEW-AHR-EYE) is the way you identify any of those points of content, whether it be a page of text, a video or sound clip, a still or animated image, or a program. The most common form of URI is the Web page address, which is a particular form or subset of URI called a Uniform Resource Locator (URL).

Storage Account

– We assume users who want to use our webpage to predict visitor amount should have an Azure Storage account. In this way, they can use their own dataset as the input of the prediction models. Also, our prediction model can store the prediction results to their own storage accounts. This method improves the security of this prediction model, and it can guarantee customers confidentiality.



Storage Account Components

Storage Components

- **Storage Account:** All access to Azure Storage is done through a storage account. This storage account can be a **General Purpose Storage Account** or a **Blob Storage Account** which is specialized for storing objects/blobs.
- **Container:** A container provides a grouping of a set of blobs. All blobs must be in a container. An account can contain an unlimited number of containers. A container can store an unlimited number of blobs. Note that the container name must be lowercase.
- **Blob:** A file of any type and size. Azure Storage offers three types of blobs: block blobs, page blobs, and append blobs.

Upload Files

- Azure Storage Account is similar with GitHub, we cannot directly upload files to this account. We must use the third party tools like Azure Powershell or .Net studio to upload files to Blob Storage Account.
- Now, we will show the process of uploading files to storage account using Azure Powershell. All screen shot came from Powershell Command Window.

Upload Files (CON'T)

- **Log-in to Azure:** A pop-up log in window will show, and customer should log in to their Azure account.

```
PS C:\Users\CANDICEHO> Login-AzureRmAccount

Environment           : AzureCloud
Account               : candiceho1215@gmail.com
TenantId              : 96ca9cb2-8460-4c50-8b45-8facc9c832cc
SubscriptionId        : eeba4fbe-f754-4282-aa24-12442c1211b5
CurrentStorageAccount :
```

- **2. Check Azure Subscription**: It will give you the subscription information about you Azure account, such as Subscription Name, Id and State.

```
PS C:\Users\CANDICEHO> Get-AzureRmSubscription

SubscriptionName : Free Trial
SubscriptionId   : eeba4fbe-f754-4282-aa24-12442c1211b5
TenantId         : 96ca9cb2-8460-4c50-8b45-8facc9c832cc
State            : Enabled
```

Upload Files (CON' T)

– **3. Check Azure Context**: It will give where you are, and which Azure account you are connecting to. But we have not yet set which Storage Account you want to connect, so that line is empty.

```
PS C:\Users\CANDICEHO> Get-AzureRmContext

Environment           : AzureCloud
Account               : candiceho1215@gmail.com
TenantId              : 96ca9cb2-8460-4c50-8b45-8facc9c832cc
SubscriptionId        : eeba4fbe-f754-4282-aa24-12442c1211b5
CurrentStorageAccount :
```

– **4. Set Storage Account**

```
PS C:\Users\CANDICEHO> Set-AzureRmCurrentStorageAccount -ResourceGroupName "test" -StorageAccountName "customer1215"
customer1215
```

We should give the Storage Account name, and the Group your account belongs to. Now, you can see the Current Storage Account is "customer1215".

```
PS C:\Users\CANDICEHO> Get-AzureRmContext

Environment           : AzureCloud
Account               : candiceho1215@gmail.com
TenantId              : 96ca9cb2-8460-4c50-8b45-8facc9c832cc
SubscriptionId        : eeba4fbe-f754-4282-aa24-12442c1211b5
CurrentStorageAccount : customer1215
```

Upload Files (CON’T)

- **5. Set Account Parameters**
- You can create several parameters, such as “$accountName”, “$containerName”, “$storage AccessKey”, and “blobContext”. It is convenient to give a uploading files command.

```
PS C:\Users\CANDICEHO> $storageAccountName = "customer1215"
PS C:\Users\CANDICEHO> $containerName = "container1"
PS C:\Users\CANDICEHO> $storageAccountKey = "6lcGoWsDu9wTEZki0RsNHAV3fttcQwcRYKxBE7pBTvZ26T5z8N5Y9fnNGHKFXqGW8qu4smyPK+0
OcAAYJ9w4Zw=="
PS C:\Users\CANDICEHO> $blobContext = New-AzureStorageContext -StorageAccountName $storageAccountName -StorageAccountKey
 $storageAccountKey
```

- **6. Upload Files**: Use parameters created before, and give the address of file which you want to upload.

```
PS C:\Users\CANDICEHO>  Set-AzureStorageBlobContent -File d:\data\test1.csv -Container $containerName -Context $blobCont
ext -Force


ICloudBlob        : Microsoft.WindowsAzure.Storage.Blob.CloudBlockBlob
BlobType          : BlockBlob
Length            : 3992
ContentType       : application/octet-stream
LastModified      : 2016/4/28 22:49:23 +00:00
SnapshotTime      :
ContinuationToken :
Context           : Microsoft.WindowsAzure.Commands.Common.Storage.AzureStorageContext
Name              : test1.csv
```

Models Integration

- We used Azure Web Service to create Web App. For the first model, we allow users to upload their own CSV files as that model's input, and the model will generate a CSV file as the output of prediction. Thus, we used Batch Execution Web App for first model.
- For second and third model, we allow users to input each variable value, and models will give them one prediction value for the certain input. Thus, we used Request-Response Web App.

Web API

- http://www1.ece.neu.edu/~zwang3/final_project_UI/home.html
- Account Name: customer1215

AccountKey: 6lcGoWsDu9wTEZki0RsNHAV3fttcQwcRYKxBE7pBTvZ26T5z8N5Y9fnNGHKFXqGW8qu4smyPK+0OcAAYJ9w4Zw==

- Container Name: container1

THANK YOU