Package 'uberdata'

April 4, 2015

Title Uber Data Prediction

Version 0.0.0.9000

Author Adam Sullivan [aut, cre]

Maintainer Adam Sullivan <adam-sullivan@live.com>

Description This package accepts a dataset from the Uber location database and generates/predicts the time of day for a new pickup location.

Depends R (>= 3.0.2)

License Internal

LazyData true

R topics documented:

`calcTripDistance` *Distance calculation example*

Description

This function accepts a start lat/long and end lat/long and returns the Haversine distance. There is an additional option to use a straight euclidean distance (not recommended).

Usage

```
calcTripDistance(distFrame, type = "haversine")
```

Arguments

`distFrame`	a data frame that includes dropoff_lat/long, and begintrip_lat/long.

Value

dist the haversine or euclidean distance, in meters

Examples

```
timeOfDayFnc(6)
# [1] "morning"
```

`featureEngineering` *Feature Generation*

Description

Take a given data frame and produce a feature vector for each unique row.

Usage

```
featureEngineering(tripData, truncatedData, newKmeans = NULL)
```

Arguments

`tripData`	A data frame with dateTime, startLat/Long, stopLat/Long, and uid.
`truncatedData`	An option to reduce the dimensionality of the return frame.

Value

featureFrame A data frame with the equivalent features calcualted `findClusteredLocations` *findClusteredLocations*

Description

This function takes every start and end location in the Uber data set and attempts to define k clusters (using kmeans). The k clusters is defined above, and defaulted to 32.

Usage

```
findClusteredLocations(dataFrame, NUMCENTERS = 32)
```

Arguments

`dataFrame`	a data frame with the date/time (as posixct)

Value

cluster cluster locationdocu

`findDayOfWeek` *findDayOfWeek*

Description

This function takes a data frame and returns the ’day of the week’.

Usage

```
findDayOfWeek(dataFrame)
```

Arguments

`dataFrame`	a data frame with the date/time (as posixct)

Value

weekday returns the "Mon", "Tues", etc day of the week for a given date.

mlogitModel *Multinomial logistic regression*

Description

This function is another exploratory attempt at using hiearachial logistic regression. Using the endCluster's as the variable to be predicted, it's still exploratory and not to be used.

Usage

```
mlogitModel(testTrip)
```

Arguments

`testTrip`	an input of the shortened feature vector

Value

cModel The model S3 object.

multinomialHierBayesModel *Bayesian Hiearchial multinomial logistic regression*

Description

Exploratory function. No guarantee on code safety, included for demonstration. This function was my top pick for being able to model the dropoff location. It creates a list structure (one for each unique ID) as the input and predictor variable. The output is MCMC samples for the estimates for beta.

Usage

```
multinomialHierBayesModel(testTrip)
```

Arguments

`testTrip`	an input of the shortened feature vector

Value

outMCMCs The multinomial Bayesian model chains for the estimates for beta.

`naiveBayesModel` *Naive Bayes Model*

Description

This function takes in a data frame and returns back a naive bayes model.

Usage

```
naiveBayesModel(testTrip)
```

Arguments

`testTrip`	a data frame generated from the dataProcess functions.

Value

nbModel The entire S3 object for the NB model

`naiveBayesTimeAnalysis` *naiveBayesTimeAnalysis*

Description

This function takes a model and testData, and returns a confusion matrix with corresponding classification errors..

Usage

```
naiveBayesTimeAnalysis(model, testData)
```

Arguments

`model`	a naive bayes model
`testData`	the input features from the testing data set.

Value

cfMatrix The confusion matrix from the NB analysis.

`preprocessData` *Preprocessing of CSV data*

Description

Load in a dataset for training and pre-process for acceptable data inputs

Usage

```
preprocessData(fileInput)
```

Arguments

`fileName`	A filename in string format.

Value

tripData A data frame of the pre-processed csv file.

`timeOfDayFnc` *Time of Day parser*

Description

This function takes in an hour and parses it into a categorical variable.

Usage

```
timeOfDayFnc(tripFrame)
```

Arguments

`hourVal`	A stripped out single hour.

Value

catHour A category of the

Examples

```
timeOfDayFnc(6)
# [1] "morning"
```

Index