In [2]:

```python
# Built by Dr. Karthik Sekaran, Ph.D.,
```

In [3]:

```python
# Importing the necessary libraries

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [4]:

```python
# import dataset

data = pd.read_csv("train.csv")
```

In [5]:

```python
# Statistical information about numerical features

data.describe()
```

Out[5]:

	distance	weight	cost
count	38999.000000	38999.000000	38999.000000
mean	2004.061643	42.293033	80.972210
std	728.996843	75.473752	180.715055
min	400.000000	1.000000	5.000000
25%	1400.000000	13.000000	38.768742
50%	2000.000000	25.000000	46.468476
75%	2600.000000	39.000000	70.239797
max	3600.000000	500.000000	2019.734797

In [6]:

```python
# Printing top 5 rows

data.head()
```

Out[6]:

	trip	date	dayPart	exWeatherTag	originLocation	destinationLocation	distance	type	weight	packa
0	t52712528	2017-09-06	night	NaN	S4	D7	2200	expedited	50	
1	t29859381	2017-10-21	night	NaN	S8	D1	1800	NaN	12	
2	t25702332	2017-07-15	night	NaN	S9	D5	2800	NaN	1	
3	t27713405	2017-10-22	day	NaN	S9	D7	3200	NaN	50	
4	t49439220	2019-12-11	day	snow	S9	D1	2000	NaN	43	

In [7]:

```python
# Checking for any null value in the dataset

data.isnull().sum()
```

Out[7]:
```
trip                      0
date                      0
dayPart                   0
exWeatherTag          34117
originLocation            0
destinationLocation       0
distance                  0
type                  35251
weight                    0
packageType           36499
carrier                   0
cost                      0
dtype: int64
```

In [8]:
```
# Converting categorical features into encoded numerical vectors

from sklearn import preprocessing

label_encoder = preprocessing.LabelEncoder()

data['dayPart']= label_encoder.fit_transform(data['dayPart'])
data['originLocation']= label_encoder.fit_transform(data['originLocation'])
data['destinationLocation']= label_encoder.fit_transform(data['destinationLocation'])
data['carrier']= label_encoder.fit_transform(data['carrier'])
```

In [9]:
```
# Dropping features having no big importance on predicting cost for the logistics and havi

data.drop(["trip", "date", "exWeatherTag", "type", "packageType"], axis=True, inplace=True
```

In [10]:
```
# Checking whether the features doesn't contains null value after dropping few

data.isnull().sum()
```

Out[10]:
```
dayPart                0
originLocation         0
destinationLocation    0
distance               0
weight                 0
carrier                0
cost                   0
dtype: int64
```

In [11]:
```
features = data.columns
features
```

Out[11]:
```
Index(['dayPart', 'originLocation', 'destinationLocation', 'distance',
       'weight', 'carrier', 'cost'],
      dtype='object')
```

In [12]:
```
data.head()
```

Out[12]:

	dayPart	originLocation	destinationLocation	distance	weight	carrier	cost
0	1	3	6	2200	50	3	68.413152
1	1	7	0	1800	12	1	36.450649
2	1	8	4	2800	1	1	9.057939
3	0	8	6	3200	50	2	57.320087

	dayPart	originLocation	destinationLocation	distance	weight	carrier	cost
4	0	8	0	2000	43	1	77.263777

In [13]:

```
# Separating data into feature variable X and target variable y respectively

from sklearn.model_selection import train_test_split
X = data.iloc[:,:-1].values
y = data.iloc[:,-1].values

# Extracting the names of the features from data

features = data.columns

# Splitting X & y into training and testing set
X_train, X_test, y_train, y_test = train_test_split(
        X, y, train_size=0.90, random_state=50)
```

In [14]:

```
# Instantiating the prediction model - an extra-trees regressor
from sklearn.ensemble import ExtraTreesRegressor
reg = ExtraTreesRegressor(random_state=50)

# Fitting the predictino model onto the training set
reg.fit(X_train, y_train)

# Checking the model's performance on the test set
print('R2 score for the model on test set =', reg.score(X_test, y_test))
```

```
R2 score for the model on test set = 0.9976544777334635
```

In [15]:

```
!pip install lime
```

```
Requirement already satisfied: lime in c:\users\karthik-sekaran\anaconda3\lib\site-package
s (0.2.0.1)
Requirement already satisfied: scipy in c:\users\karthik-sekaran\anaconda3\lib\site-packag
es (from lime) (1.7.1)
Requirement already satisfied: scikit-image>=0.12 in c:\users\karthik-sekaran\anaconda3\li
b\site-packages (from lime) (0.18.3)
Requirement already satisfied: scikit-learn>=0.18 in c:\users\karthik-sekaran\anaconda3\li
b\site-packages (from lime) (0.24.2)
Requirement already satisfied: numpy in c:\users\karthik-sekaran\anaconda3\lib\site-packag
es (from lime) (1.20.3)
Requirement already satisfied: matplotlib in c:\users\karthik-sekaran\anaconda3\lib\site-p
ackages (from lime) (3.4.3)
Requirement already satisfied: tqdm in c:\users\karthik-sekaran\anaconda3\lib\site-package
s (from lime) (4.62.3)
Requirement already satisfied: networkx>=2.0 in c:\users\karthik-sekaran\anaconda3\lib\sit
e-packages (from scikit-image>=0.12->lime) (2.6.3)
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,>=4.3.0 in c:\users\karthik-sekaran\a
naconda3\lib\site-packages (from scikit-image>=0.12->lime) (8.4.0)
Requirement already satisfied: imageio>=2.3.0 in c:\users\karthik-sekaran\anaconda3\lib\si
te-packages (from scikit-image>=0.12->lime) (2.9.0)
Requirement already satisfied: tifffile>=2019.7.26 in c:\users\karthik-sekaran\anaconda3\l
ib\site-packages (from scikit-image>=0.12->lime) (2021.7.2)
Requirement already satisfied: PyWavelets>=1.1.1 in c:\users\karthik-sekaran\anaconda3\lib
\site-packages (from scikit-image>=0.12->lime) (1.1.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\karthik-sekaran\anaconda3
\lib\site-packages (from matplotlib->lime) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\karthik-sekaran\anaconda3\lib
\site-packages (from matplotlib->lime) (3.0.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\karthik-sekaran\anaconda3\lib
\site-packages (from matplotlib->lime) (1.3.1)
```

```
Requirement already satisfied: cycler>=0.10 in c:\users\karthik-sekaran\anaconda3\lib\site
-packages (from matplotlib->lime) (0.10.0)
Requirement already satisfied: six in c:\users\karthik-sekaran\anaconda3\lib\site-packages
(from cycler>=0.10->matplotlib->lime) (1.16.0)
Requirement already satisfied: joblib>=0.11 in c:\users\karthik-sekaran\anaconda3\lib\site
-packages (from scikit-learn>=0.18->lime) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\karthik-sekaran\anaconda3
\lib\site-packages (from scikit-learn>=0.18->lime) (2.2.0)
Requirement already satisfied: colorama in c:\users\karthik-sekaran\anaconda3\lib\site-pac
kages (from tqdm->lime) (0.4.4)
```

In [16]:

```python
# Importing the module for LimeTabularExplainer
import lime.lime_tabular

# Instantiating the explainer object by passing in the training set, and the extracted fea
explainer_lime = lime.lime_tabular.LimeTabularExplainer(X_train,
```

In [17]:

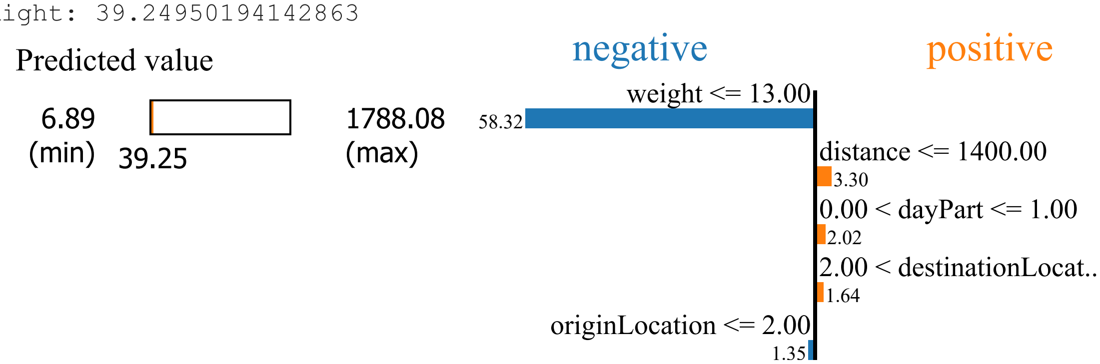
```python
# Index corresponding to the test vector
i = 100

# Number denoting the top features
k = 5

# Calling the explain_instance method by passing in the:
# 1) ith test vector
# 2) prediction function used by our prediction model('reg' in this case)
# 3) the top features which we want to see, denoted by k
exp_lime = explainer_lime.explain_instance(
        X_test[i], reg.predict, num_features=k)

# Finally visualizing the explanations
exp_lime.show_in_notebook()
print("Actual Prediction Score:", y_test[i])
```

```
Intercept 88.6209770174844
Prediction_local [35.91077449]
Right: 39.24950194142863
```



Feature	Value
weight	13.00
distance	1000.00
dayPart	1.00
destinationLocation	3.00
originLocation	0.00

```
Actual Prediction Score: 40.56659932
```

In [18]:

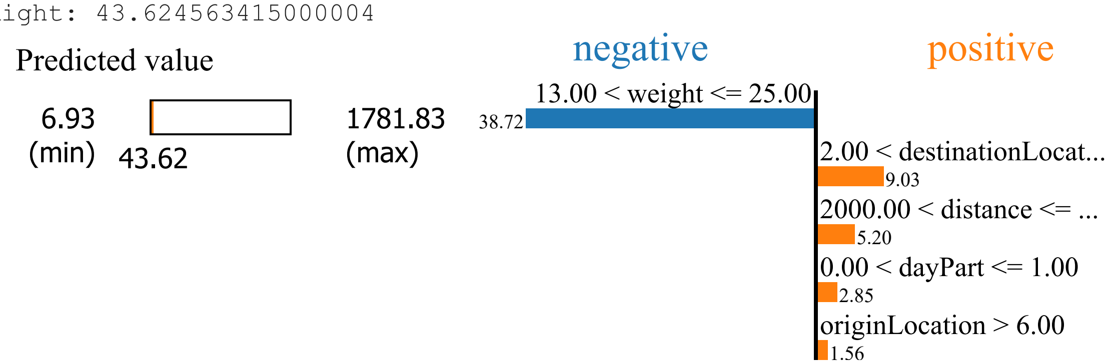
```python
# Index corresponding to the test vector
i = 47

# Number denoting the top features
k = 5

# Calling the explain_instance method by passing in the:
# 1) ith test vector
# 2) prediction function used by our prediction model('reg' in this case)
# 3) the top features which we want to see, denoted by k
exp_lime = explainer_lime.explain_instance(
        X_test[i], reg.predict, num_features=k)

# Finally visualizing the explanations
exp_lime.show_in_notebook()
```

```
Intercept 75.12463039813042
Prediction_local [55.03300346]
Right: 43.624563415000004
```



Feature	Value
weight	23.00
destinationLocation	3.00
distance	2600.00
dayPart	1.00
originLocation	8.00