Motivation

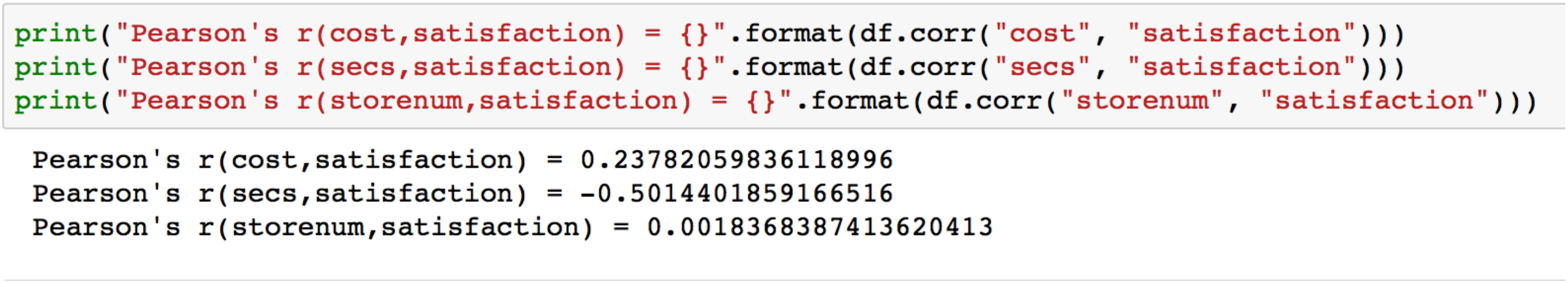
- The dataset of fast food contains several variables including cost of food and customers' satisfaction rating, which are the regressor and response variable that I want to build model on and test their relationship.

- Using MLlib based analysis to check the Pearson's ratio for regressors to see whether they have strong relationships with response variable.

- Making training set and test set, building a linear regression model and transforming the test set to get predictions.

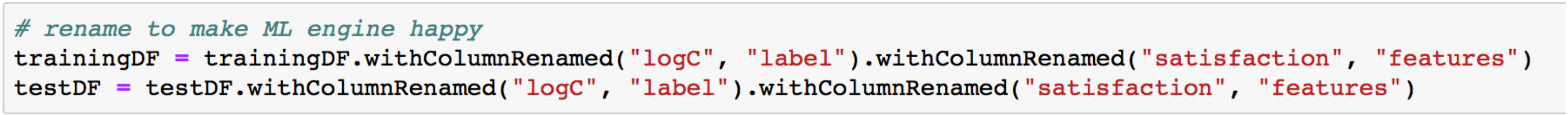
- Finally make a model evaluation to test the accuracy and use data visualization to interpret the relationship between regressor and response variable

Code Snippet and Explanation

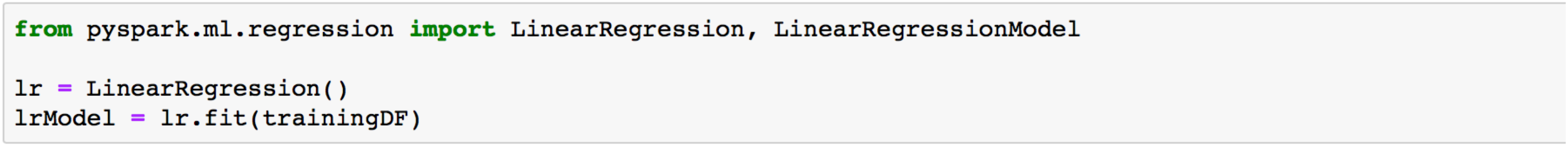


```
print("Pearson's r(cost,satisfaction) = {}".format(df.corr("cost", "satisfaction")))
print("Pearson's r(secs,satisfaction) = {}".format(df.corr("secs", "satisfaction")))
print("Pearson's r(storenum,satisfaction) = {}".format(df.corr("storenum", "satisfaction")))
```

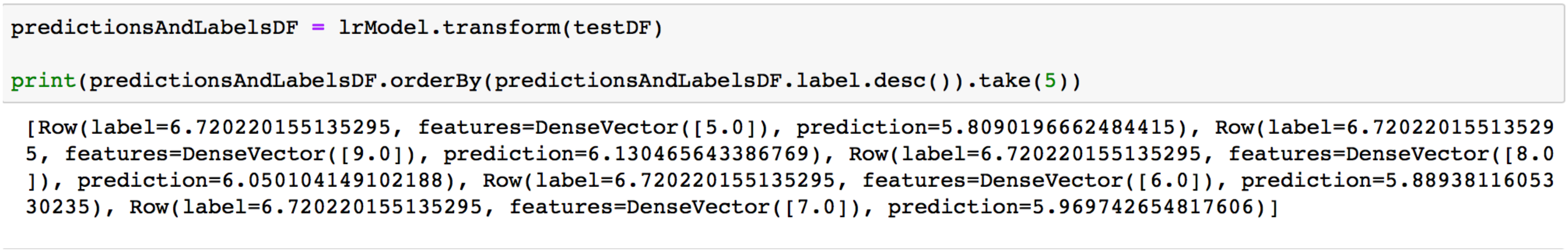
```
Pearson's r(cost,satisfaction) = 0.23782059836118996
Pearson's r(secs,satisfaction) = -0.5014401859166516
Pearson's r(storenum,satisfaction) = 0.0018368387413620413
```



```
# rename to make ML engine happy
trainingDF = trainingDF.withColumnRenamed("logC", "label").withColumnRenamed("satisfaction", "features")
testDF = testDF.withColumnRenamed("logC", "label").withColumnRenamed("satisfaction", "features")
```

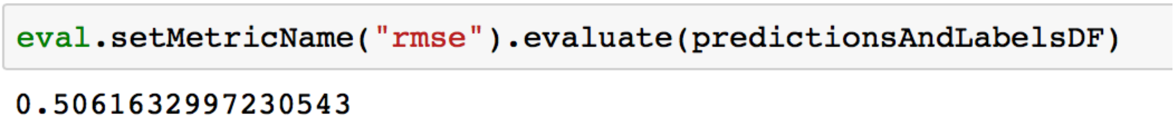


```
from pyspark.ml.regression import LinearRegression, LinearRegressionModel

lr = LinearRegression()
lrModel = lr.fit(trainingDF)
```



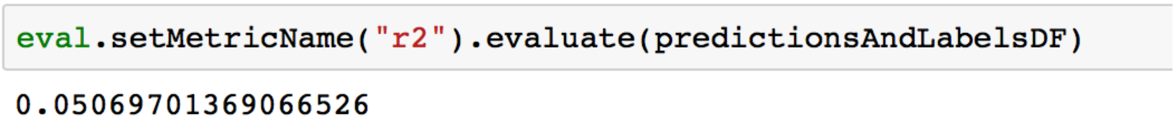
```
predictionsAndLabelsDF = lrModel.transform(testDF)

print(predictionsAndLabelsDF.orderBy(predictionsAndLabelsDF.label.desc()).take(5))
```

```
[Row(label=6.720220155135295, features=DenseVector([5.0]), prediction=5.8090196662484415), Row(label=6.72022015513529
5, features=DenseVector([9.0]), prediction=6.130465643386769), Row(label=6.720220155135295, features=DenseVector([8.0
]), prediction=6.050104149102188), Row(label=6.720220155135295, features=DenseVector([6.0]), prediction=5.88938116053
30235), Row(label=6.720220155135295, features=DenseVector([7.0]), prediction=5.969742654817606)]
```



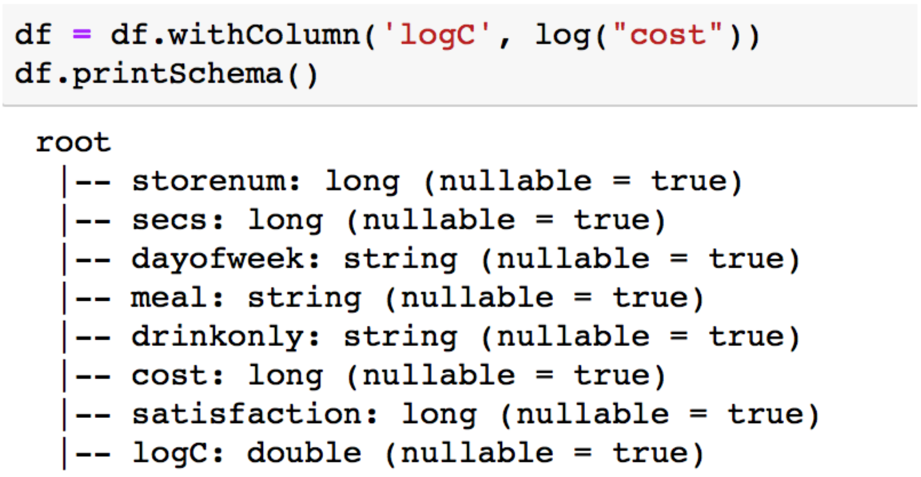
```
eval.setMetricName("rmse").evaluate(predictionsAndLabelsDF)
```

```
0.5061632997230543
```



```
eval.setMetricName("r2").evaluate(predictionsAndLabelsDF)
```

```
0.05069701369066526
```



```
df = df.withColumn('logC', log("cost"))
df.printSchema()
```

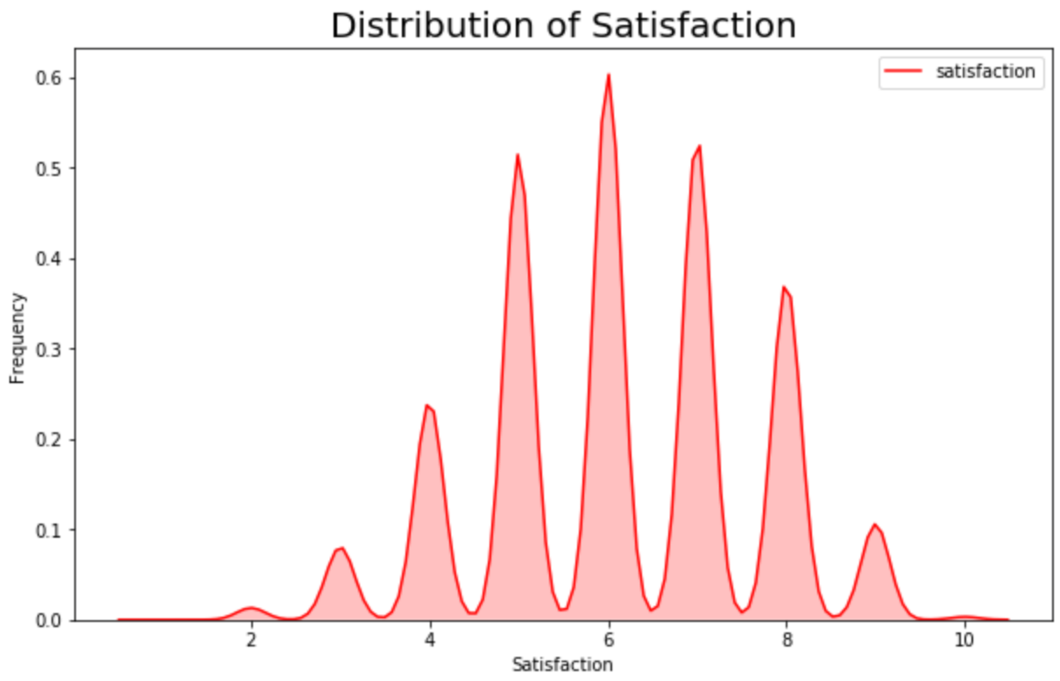
```
root
 |-- storenum: long (nullable = true)
 |-- secs: long (nullable = true)
 |-- dayofweek: string (nullable = true)
 |-- meal: string (nullable = true)
 |-- drinkonly: string (nullable = true)
 |-- cost: long (nullable = true)
 |-- satisfaction: long (nullable = true)
 |-- logC: double (nullable = true)
```

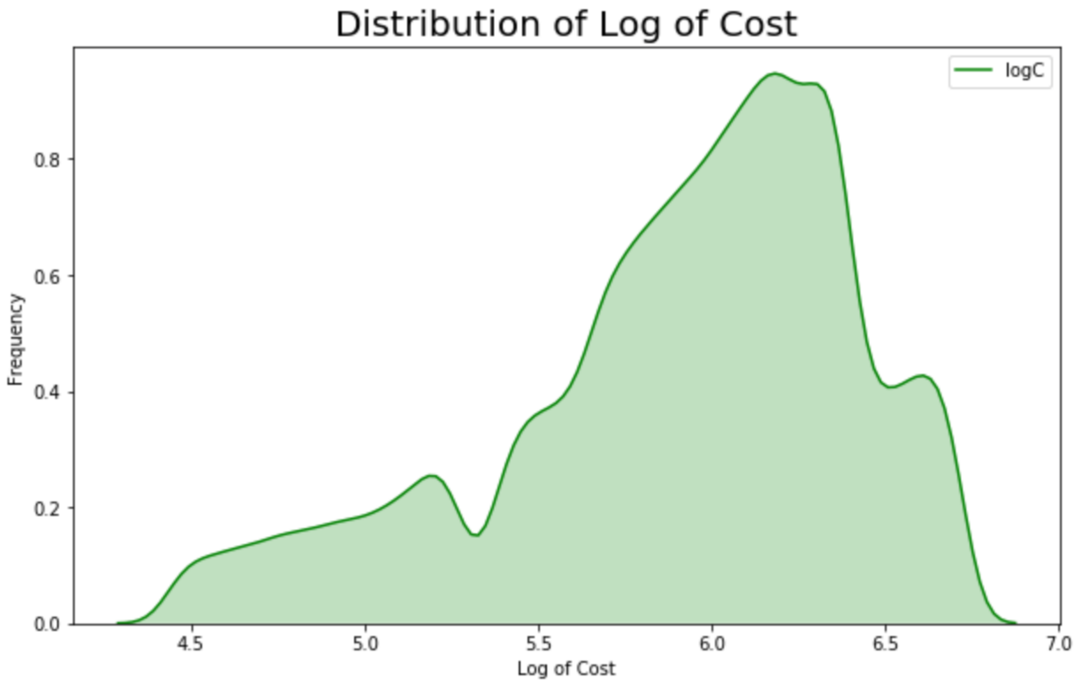
- **By using MLlib based analysis to check the Pearson’s ratio for three regressors, since all of the Pearson’s ratio is not large enough to identify an relationship, I use the log transformation of cost.**

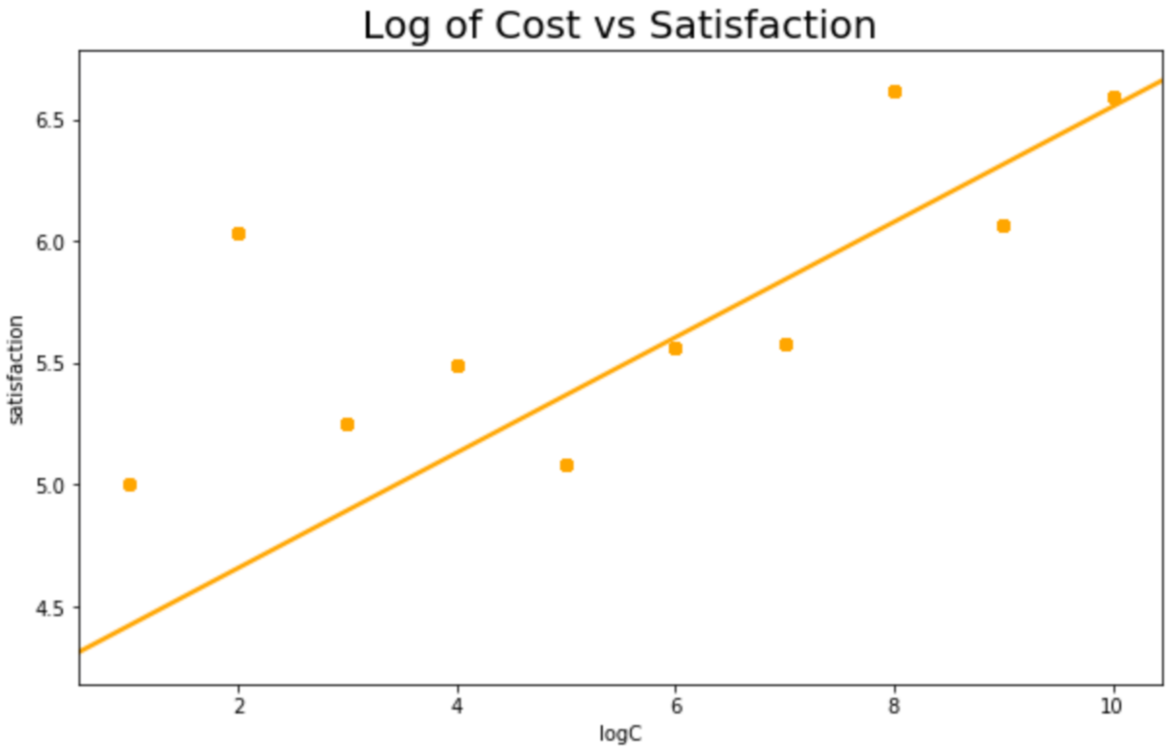
- **Making training set and test set, building a linear regression model and transforming the test set to get predictions.**

- **Finally make a model evaluation to test the accuracy and use data visualization to interpret the relationship between cost and satisfaction.**

Data Visualization


Distribution of Satisfaction
satisfaction
Frequency
Satisfaction


Distribution of Log of Cost
logC
Frequency
Log of Cost


Log of Cost vs Satisfaction
satisfaction
logC