# Consulting Project

## Recommender Systems

The whole world seems to be hearing about your new amazing abilities to analyze big data and build useful systems for them! You've just taken up a new contract with a new online food delivery company. This company is trying to differentiate itself by recommending new meals to customers based off of other customers likings.

Can you build them a recommendation system?

Your final result should be in the form of a function that can take in a Spark DataFrame of a single customer's ratings for various meals and output their top 3 suggested meals. For example:

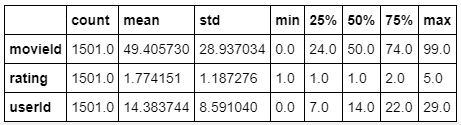
Best of luck!

**Note from Jose: I completely made up this food data, so its likely that the actual recommendations themselves won't make any sense. But you should get a similar output to what I did given the example customer dataframe**

**import** **pandas** **as** **pd**

df = pd.read\_csv('movielens\_ratings.csv')

df.describe().transpose()



df.corr()

**import** **numpy** **as** **np**

df['mealskew'] = df['movieId'].apply(**lambda** id: np.nan **if** id > 31 **else** id)

df.describe().transpose()

df['meal\_name'] = df['mealskew'].map(mealmap)

df.to\_csv('Meal\_Info.csv',index=**False**)

**from** **pyspark.sql** **import** SparkSession

spark = SparkSession.builder.appName('recconsulting').getOrCreate()

**from** **pyspark.ml.evaluation** **import** RegressionEvaluator

**from** **pyspark.ml.recommendation** **import** ALS

data = spark.read.csv('Meal\_Info.csv', inferSchema=**True**, header=**True**)

(training, test) = data.randomSplit([0.8, 0.2])

*# Build the recommendation model using ALS on the training data*

als = ALS(maxIter=5, regParam=0.01, userCol="userId", itemCol="mealskew", ratingCol="rating")

model = als.fit(training)

*# Evaluate the model by computing the RMSE on the test data*

predictions = model.transform(test)

predictions.show()

evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating",predictionCol="prediction")

rmse = evaluator.evaluate(predictions)

print("Root-mean-square error = " + str(rmse))