# Wait Times for Disney World Attractions

Team members: Reema Singh, Jessika Virtanen, Kimberly Gordon, Melissa Wood, Serena Terry, Sean McCray

Datasets being used: <https://touringplans.com/walt-disney-world/crowd-calendar#DataSets>

Overview: Use machine learning to see if we can predict wait times at Walt Disney World attractions, create visualizations looking at wait times and attraction popularity during various times of year, and create a map showing attraction information.

Project utilizes: ML, HTML/CSS/Bootstrap, Tableau, Pandas, Javascript

More detailed description of the ML portion:

**Machine Learning:** The idea was to use ML to run through the data with dates and wait times for over 5 years to predict the wait times for various rides

Date column was converted into meaningful features such as -> 'year', 'month', 'day', 'dayofweek', 'dayofyear', 'weekofyear', 'quarter','is\_month\_start','is\_month\_end','is\_quarter\_start'

We determined if the date was Holiday and if Yes then which holiday (New Year, Christmas, Independence day ..)? Data was scaled and divided into train and test. Features and Y were HOT encoded as needed

More than 3 different approaches were utilized for ML

Linear regression with wait times as bins

Weight coefficients: [[3.80478216e-05]]

y-axis intercept: [10.44120957]

**Deep Neural Network:**  Sequential Model with 25  inputs, 2 hidden nodes, and 20 outputs run for 1000 epocs. Wait times as bins

model.compile(optimizer='adam',

              loss='categorical\_crossentropy',

              metrics=['accuracy'])

Final Result : loss: 0.9444 - acc: 0.4342

**Deep Neural Network:**  Sequential Model with 25  inputs, 3 hidden nodes, and 1 outputs run for 1000 epocs. Wait time as time

model.compile(optimizer="adam", loss="mean\_squared\_error")

model = Sequential()

model.add(Dense(units=1000, activation='relu', input\_dim=25))

model.add(Dense(units=100, activation='relu'))

model.add(Dense(units=200, activation='relu'))

model.add(Dense(units=1, activation='linear'))

np.sqrt(model.evaluate(X\_test, y\_test))=6.4737539257165935

**Predictions:** Input date is scaled and all converted to have all the features as train data. Saved models were uploaded and used for predictions