Covariate Shift Tool Detection

Problem Statement

Use Case

Summary of Tool

Psuedocode

During Training

```
From Train Data: Split each feature into separate vector

###

  # Example:
  # Train data:
  # x_1 = height
  # x_2 = weight
  # x_3 = gender
  # y = average calories consumed
  # Separate vector for each feature
  # x_1 = [x_1_1, x_2_1, ... , x_n_1]
  # x_2 = [x_1_2, x_2_2, ... , x_n_2]
  # etc.

###

For each feature vector i:

  # Kernel Density estimate just for feature vector i

  kde_i <= Run Kernel Density Estimation algorithm on i

  # Multivariate Kernel Density estimate for feature vector i and label y
vector

  kde_mv_i <= Run Multivariate Kernel Density Estimation algorithm on i
```

In Production - Upon publication

```
Upon publication of model to production:
  Create blank feature 2-D vectors for each (feature, date)

  # Example:
  # x_prod_1 = (height, prediction_ID, date)
  # x_prod_2 = (weight, prediction_ID, date)
  # x_prod_3 = (gender, prediction_ID, date)
  # y_prod = (average calories consumed, prediction_ID, date)
```

In Production - Upon new prediction request

```
Upon new prediction request to platform:
  Send data to covariate shift tool function

  Upon receipt of data at covariate shift tool:
    Separate input data into individual elements

    Generate prediction_ID

    # Example:
    # Input data: {{height: 73}, {weight: 160}, {gender: 1}}
    # x_1 = (73, prediction_ID, current_date)
    # x_2 = (160, prediction_ID, current_date)
    # x_3 = (1, prediction_ID, current_date)

    Append these elements to respective 2-D vectors

    # Example:
    # x_prod_1 << x_1
    # x_prod_2 << x_2
    # x_prod_3 << x_3

    Submit input data to model for prediction and wait for result

    Receive result and create output element

    # Example:
    # Model predicts 2100
    # y = (2100, prediction_ID, current_date)

    Append output element to respective vector

    # Example:
    # y_prod << y
```

In Production - Visualize covariate shift 1-D

```
Select feature i to analyze covariate shift on

# Kernel Density estimate just for production feature vector i

kde_prod_i <= Run Kernel Density Estimation algorithm on i (potentially
randomly sample subset of dataset)

Graph kde_mi and kde_prod_i
```

In Production - Visualize covariate shift 2-D

```
Select feature i to analyze covariate shift on

# Multivariate Kernel Density estimate just for production feature i and label
y vector

kde_prod_mv_i <= Run Kernel Density Estimation algorithm on i and associated
labels y (potentially randomly sample subset of dataset)

Graph kde_mv_i and kde_prod_mv_i
```

In Production - Automatic detection of covariate shift

```
For every N number of prediction requests // must decide what N is?:
  train_data = Take random X samples of training data and label them training
  production_data = Take random X samples of production data and label them
production

  full_data = train_data + production_data



  Train a logistic regression model (for speed) to classify train vs production
data
```