

Documento Técnico

Metodología:

Nuestra solución consiste en realizar una regresión lineal sobre los minutos acumulados sobre la temperatura (263 C) para poder predecir en qué ciclo el motor necesita una reparación dependiendo de sus minutos acumulados. Además predecimos la probabilidad en que ciudad el motor se va a dañar junto con los costos asociados de hacer la reparación en esa ciudad.

Para poder lograr esto realizamos una limpieza inicial donde sustituimos todos los tiempos negativos y los valores "NA" por el promedio correspondiente de la columna, ya sea temperatura del motor y minutos a esa temperatura.

Una vez hecha esta limpieza se generó un script en Python que procesa los datos y nos regresa un archivo CSV donde conseguimos los datos de cada ecuación de regresión para cada motor, como también la ruta que le corresponde más la ciudad en la que se encuentra.

MicroServicios:

- enginesList(): Este microservicio se encarga de generar la lista de motores para el selector de dropdown.
- engines(esn): Este servicio se encarga de generar los valores para definir la tendencia para cada motor, para los siguientes 50 ciclos.
- engineTable(esn): Este servicio genera las probabilidades acumuladas de que en el siguiente ciclo se dañe el motor en una ciudad específica, este genera la probabilidad para 5000 ciclos.
- critical(): Genera la lista de engines que se encuentran en un estado crítico y necesitan reparación en un momento cercano.

FrontEnd:

- Engine failure chance: En esta vista desplegamos una tabla de probabilidades en donde se selecciona un motor y podemos ver que probabilidad existe para que un motor se

daño en cada ciudad en específico y el costo que representa reparar el motor en esa ciudad.

```
define(['angular', './sample-module'], function(angular, sampleModule) {
'use strict';
return sampleModule.controller('SampleCtrl', ['$scope', '$http', function($scope, $http) {
  function fillEnginesDropdown($scope,$http) {
    $http({
      method: 'GET',
      url: 'http://localhost:12121/engines/list',
      headers: {

    }
    }).
    success(function(data){
      $scope.engines = data;
      console.log($scope.data);
      console.log('Successful call');
    });
  };

  function getDataForEngine($scope,$http,engineNumber) {
    $http({
      method: 'GET',
      url: 'http://localhost:12121/engineTable/' + engineNumber,
      headers: {

    }
    }).
    success(function(data){
      $scope.engineData = data;
      console.log($scope.engineData);
      console.log('Successful call');
    });
  };

  fillEnginesDropdown($scope,$http);
  getDataForEngine($scope,$http,700101);
  document.getElementById('enginesDropdown').addEventListener('dropdown_content_value_changed',
function(e) {
    var selectedEngine = e.detail;
    $scope.selectedEngine = selectedEngine.textValue;
    console.log($scope.currentESN);
    $scope.engineData = [];
    getDataForEngine($scope, $http, $scope.selectedEngine);
  });

}]);
```

```
});
```

- Engine temperatures: Esta visualización muestra la línea de tendencia de un motor y el threshold en el cual es lo más probable que se dañe el motor. También se puede seleccionar el motor con un dropdown.

```
define(['angular', './sample-module'], function(angular, sampleModule) {
  'use strict';
  return sampleModule.controller('SampleCtrl2', ['$scope', '$http', function($scope, $http) {
    function fillEnginesDropdown($scope,$http) {
      $http({
        method: 'GET',
        url: 'http://localhost:12121/engines/list',
        headers: {

        }
      }).
      success(function(data){
        $scope.engines = data;
        console.log($scope.data);
        console.log('Successful call');
      });
    };

    function getDataForEngine($scope,$http,engineNumber) {
      $http({
        method: 'GET',
        url: 'http://localhost:12121/engines/' + engineNumber,
        headers: {

        }
      }).
      success(function(data){
        $scope.minY = 9999999;
        $scope.maxY = 0;
        data.forEach(getMinAndMax);
        $scope.maxY = Math.round($scope.maxY + 1000)*100/100 ;
        $scope.minY = Math.round($scope.minY- 1000)*100/100 ;
        if ($scope.minY < 0){
          $scope.minY = 0;
        }
        $scope.engineData = data;
        console.log($scope.data);
        console.log('Successful call');
      });
    };

    fillEnginesDropdown($scope,$http);
    getDataForEngine($scope,$http,700101);
```

```

document.getElementById('enginesDropdown').addEventListener('dropdown_content_value_changed',
function(e) {
    var selectedEngine = e.detail;
    $scope.selectedEngine = selectedEngine.textValue;
    console.log($scope.currentESN);
    getDataForEngine($scope, $http, $scope.selectedEngine);
});
function getMinAndMax(element, index, array) {
    //console.log(element[1]);
    if (element[1] > $scope.maxY){
        $scope.maxY = element[1];
    }
    if (element[1] < $scope.minY){
        $scope.minY = element[1];
    }
}
}
});
});

```

- Critical engines: Esta vista muestra una lista de motores que se encuentran arriba de los 11 mil minutos y que necesitan ser reparados pronto.

```

define(['angular', './sample-module'], function(angular, sampleModule) {
    'use strict';
    return sampleModule.controller('SampleCtrl3', ['$scope', '$http', function($scope, $http) {

        function getCriticalEngines($scope,$http,engineNumber) {
            $http({
                method: 'GET',
                url: 'http://localhost:12121/engines/critical',
                headers: {

                }
            }).
            success(function(data){
                $scope.engineData = data;
                console.log($scope.engineData);
            });
        };
        getCriticalEngines($scope,$http);
    }]);
});

```