



```
logger.py  main.py  training_Validation_Insertion.py  schema_training.json  rawValidation.py  DataTypeValidation.py  DataTransformation.py  trai...
Users > aniketsatishjoshi > Download... > ML PROJECTS FSDS NOV-21 > 4 # Wafer_Fault_Detection > code > WaferFaultDetection_new > predictFromModel.py > prediction
1  import pandas
2  from file_operations import file_methods
3  from data_preprocessing import data_preprocessor
4  from data_ingestion import data_loader_prediction
5  from application_logging import logger
6  from Prediction_Raw_Data_Validation.predictionDataValidation import Prediction_Data_validation
7
8
9  class prediction:
10
11      def __init__(self,path):
12          self.file_object = open("Prediction_Logs/Prediction_Log.txt", 'a+')
13          self.log_writer = logger.App_Logger()
14          if path is not None:
15              self.pred_data_val = Prediction_Data_validation(path)
16
17      def predictionFromModel(self):
18
19          try:
20              self.pred_data_val.deletePredictionFile() #deletes the existing prediction file from last run!
21              self.log_writer.log(self.file_object,'Start of Prediction')
22              data_getter=data_loader_prediction.Data_Getter_Pred(self.file_object,self.log_writer)
23              data=data_getter.get_data()
24
25              #code change
26              # wafer_names=data['Wafer']
27              # data=data.drop(labels=['Wafer'],axis=1)
28
29              preprocessor=preprocessing.Preprocessor(self.file_object,self.log_writer)
30              is_null_present=preprocessor.is_null_present(data)
31              if(is_null_present):
32                  data=preprocessor.impute_missing_values(data)
33
34              cols_to_drop=preprocessor.get_columns_with_zero_std_deviation(data)
35              data=preprocessor.remove_columns(data,cols_to_drop)
36              #data=data.to_numpy()
37              file_loader=file_methods.File_Operation(self.file_object,self.log_writer)
38              kmeans=file_loader.load_model('KMeans')
39
40              ##Code changed
41              #pred_data = data.drop(['Wafer'],axis=1)
42              clusters=kmeans.predict(data.drop(['Wafer'],axis=1))#drops the first column for cluster prediction
43              data['clusters']=clusters
44              clusters=data['clusters'].unique()
```




