Code

Main.java

```
package main.java.sample;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.image.Image;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

public class Main extends Application {
    double x;
    double y = 0.0D;

    public Main() {
    }

    public void start(Stage stage) throws Exception {
        Parent root =
(Parent)FXMLLoader.load(this.getClass().getResource("views/dash.fxml"));
        stage.setTitle("Covid Classifier");
        stage.initStyle(StageStyle.TRANSPARENT);
        Image image = new Image("main/java/imgs/family.png");
        root.setOnMousePressed((event) -> {
            this.x = event.getSceneX();
            this.y = event.getSceneY();
        });
        root.setOnMouseDragged((event) -> {
            stage.setX(event.getScreenX() - this.x);
            stage.setY(event.getScreenY() - this.y);
        });
        stage.setScene(new Scene(root));
        stage.getIcons().add(image);
        stage.show();
    }

    public static void main(String[] args) {
        launch(args);
```

```
    }
}
```

DashController.java

```
package main.java.sample.controllers;
// Source code recreated from a .class file by IntelliJ IDEA
// (powered by FernFlower decompiler)
//

import java.lang.Thread;

import javafx.application.Platform;
import  javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.Node;
import javafx.scene.chart.BarChart;
import javafx.scene.chart.LineChart;
import javafx.scene.chart.PieChart;
import javafx.scene.chart.PieChart.Data;
import javafx.scene.chart.XYChart;
import javafx.scene.control.Label;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Pane;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Line;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import opennlp.tools.doccat.DoccatModel;
import opennlp.tools.doccat.DocumentCategorizer;
import opennlp.tools.doccat.DocumentCategorizerME;
import opennlp.tools.namefind.NameFinderME;
import opennlp.tools.namefind.TokenNameFinderModel;
import opennlp.tools.tokenize.TokenizerME;
import opennlp.tools.tokenize.TokenizerModel;
import opennlp.tools.util.Span;
import org.apache.commons.csv.CSVFormat;
import org.apache.commons.csv.CSVRecord;
```

```
import java.io.*;
import java.net.URI;
import java.net.URL;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.util.Random;
import java.util.ResourceBundle;


public class DashController implements Initializable {
    public static String virusUrl1 =
"https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_
confirmed_global.csv";
    public static String virusUrl3 =
"https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_
deaths_global.csv";
    public static String virusUrl2 =
"https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_
recovered_global.csv";

    static String[] anaInput;
    static double probc1,probc2,probsc1=0.0,probsc2,probsc3,probsc4;
    static boolean flag=false;

    @FXML
    private PieChart
pieSub,analysisMEC,analysisMESC,analysisPC,analysisPSC;
    @FXML
    private LineChart<String, Number> lineSub;
    @FXML
    private Label lblClose;
    @FXML
    private Label lblExpired, locations, dates, names,
analysisl1,analysisl2, analysisl3, analysisl4;
    @FXML
    public TextArea namespane1, datespane1, locationspane1, namespane2,
datespane2, locationspane2;
    @FXML
    private Label lbl7;
    @FXML
```

```
    private Label lbl30;
    @FXML
    private Label lbl90;
    @FXML
    private Label lblExp;
    @FXML
    private Label lblActive;
    @FXML
    private Label lblOffline;
    @FXML
    private Text pietext;
    @FXML
    private Line line1, line2;
    @FXML
    private Pane pane1, pane2, pane3, pane4, pane5;
    @FXML
    private Circle shpActive;
    @FXML
    private TextField maintext, maintext1;

    @FXML
    private TextArea category, subCategory;

    Random r = new Random();

    public DashController() {

    }

    public void initialize(URL location, ResourceBundle resources) {
        int ct = 100,rt=90,dt=10,c1 = 150,r1=130,d1=30,c2 =
250,r2=200,d2=50,c3 = 350,r3=290,d3=100;
        HttpClient client = HttpClient.newHttpClient();
        HttpRequest request1 =
HttpRequest.newBuilder().uri(URI.create(virusUrl1)).build();
        try {
            HttpResponse<String> httpResponse = client.send(request1,
HttpResponse.BodyHandlers.ofString());
            StringReader csvBodyReader = new
StringReader(httpResponse.body());
            Iterable<CSVRecord> records =
CSVFormat.DEFAULT.withFirstRecordAsHeader().parse(csvBodyReader);
            for (CSVRecord record : records) {
                //System.out.println(record.get("Country/Region"));
```

```
                if(record.get("Country/Region").equals("India")) {
                    String confirmedCases = (record.get(record.size()-10));
                    System.out.println("India" + confirmedCases);
                    lblActive.setText(confirmedCases + " confirmed");
                    ct=Integer.parseInt(confirmedCases);
                    c1=Integer.parseInt(record.get(record.size()-31));
                    c2=Integer.parseInt(record.get(record.size()-62));
                    c3=Integer.parseInt(record.get(record.size()-90));
                    System.out.println(ct+" "+c1+" "+c2);
                }
            }
        } catch(IOException e) {
            e.printStackTrace();
        } catch(InterruptedException e) {
            e.printStackTrace();
        }
        HttpRequest request2 =
HttpRequest.newBuilder().uri(URI.create(virusUrl2)).build();
        try {
            HttpResponse<String> httpResponse = client.send(request2,
HttpResponse.BodyHandlers.ofString());
            StringReader csvBodyReader = new
StringReader(httpResponse.body());
            Iterable<CSVRecord> records =
CSVFormat.DEFAULT.withFirstRecordAsHeader().parse(csvBodyReader);
            for (CSVRecord record : records) {
                //System.out.println(record.get("Country/Region"));
                if(record.get("Country/Region").equals("India")) {
                    String recoveredCases = (record.get(record.size()-10));
                    System.out.println("India" + recoveredCases);
                    lblOffline.setText(recoveredCases + " recovered");
                    rt=Integer.parseInt(recoveredCases);
                    r1=Integer.parseInt(record.get(record.size()-31));
                    r2=Integer.parseInt(record.get(record.size()-62));
                    r3=Integer.parseInt(record.get(record.size()-90));
                    System.out.println(rt+" "+r1+" "+r2);
                }
            }
        } catch(IOException e) {
            e.printStackTrace();
        } catch(InterruptedException e) {
            e.printStackTrace();
        }
        HttpRequest request3 =
```

```
HttpRequest.newBuilder().uri(URI.create(virusUrl3)).build();
        try {
            HttpResponse<String> httpResponse = client.send(request3,
HttpResponse.BodyHandlers.ofString());
            StringReader csvBodyReader = new
StringReader(httpResponse.body());
            Iterable<CSVRecord> records =
CSVFormat.DEFAULT.withFirstRecordAsHeader().parse(csvBodyReader);
            for (CSVRecord record : records) {
                //System.out.println(record.get("Country/Region"));
                if(record.get("Country/Region").equals("India")) {
                    String deceasedCases = (record.get(record.size()-10));
                    System.out.println("India" + deceasedCases);
                    lblExp.setText(deceasedCases + " expired");
                    dt=Integer.parseInt(deceasedCases);
                    d1=Integer.parseInt(record.get(record.size()-31));
                    d2=Integer.parseInt(record.get(record.size()-62));
                    d3=Integer.parseInt(record.get(record.size()-90));
                    System.out.println(dt+" "+d1+" "+d2);
                }
            }
        } catch(IOException e) {
            e.printStackTrace();
        } catch(InterruptedException e) {
            e.printStackTrace();
        }


        XYChart.Series<String,Number> series2 = new XYChart.Series<>();
        series2.getData().add(new XYChart.Data<>("2 months back",(c2-
c3)/100));
        series2.getData().add(new XYChart.Data<>("1 month back",(c1-
c2)/100));
        series2.getData().add(new XYChart.Data<>("This month",(ct-
c1)/100));
        series2.setName("Confirmed");

        XYChart.Series<String,Number> series3 = new XYChart.Series<>();
        series3.getData().add(new XYChart.Data<>("2 months back",(r2-
r3)/100));
        series3.getData().add(new XYChart.Data<>("1 month back",(r1-
r2)/100));
        series3.getData().add(new XYChart.Data<>("This month ",(rt-
r1)/100));
```

```
        series3.setName("Recovered");

        XYChart.Series<String,Number> series= new XYChart.Series<>();
        series.getData().add(new XYChart.Data<>("2 months back",(d2-d3)));
        series.getData().add(new XYChart.Data<>("1 month back",(d1-d2)));
        series.getData().add(new XYChart.Data<>("This month",(dt-d1)));
        series.setName("Deaths");

        lineSub.getData().addAll(series,series2,series3);
    }


    public void loadchart(PieChart P,double c,double r,double o,double s)
    {
        line1.setVisible(false);
        line2.setVisible(false);
        names.setVisible(false);
        dates.setVisible(false);
        locations.setVisible(false);
        pietext.setVisible(false);
        P.setVisible(true);

        ObservableList<PieChart.Data>
list=FXCollections.observableArrayList(
                new PieChart.Data("Statistics",s),
                new PieChart.Data("Corona",c),
                new PieChart.Data("Requests",r),
                new PieChart.Data("Offers",o)
        );
        P.setData(list);
        for(PieChart.Data data: P.getData())
        {
            int val= (int) (data.getPieValue()*10000);
            float v=(float)val/100;
            data.nameProperty().set(data.getName()+": "+v);
        }

    }

    public void loadAnalysisChart(PieChart p,double c, double o)
    {
        ObservableList<PieChart.Data>
list=FXCollections.observableArrayList(
                new PieChart.Data("Corona",c),
```

```
                new PieChart.Data("Others",o)
        );
        p.setData(list);
        for(PieChart.Data data: p.getData())
        {
            int val= (int) (data.getPieValue()*10000);
            float v=(float)val/100;
            data.nameProperty().set(data.getName()+": "+v);
        }
    }

    public void close(MouseEvent mouseEvent)
    {
        System.exit(0);
    }
    public void min(MouseEvent mouseEvent) {
        Stage s=(Stage)
((Node)mouseEvent.getSource()).getScene().getWindow();
        s.setIconified(true);
    }

    public void handle(ActionEvent actionEvent) {
        try {
            //Load Model
            analysisl1.setVisible(false);
            analysisl2.setVisible(false);

            analysisPSC.getData().clear();
            analysisMESC.getData().clear();
            analysisPC.getData().clear();
            analysisPSC.getData().clear();

            //////new thread with lambda/////
            Runnable analysisPage= () -> {
                    InputStream modelIn = null;
                    try {
                        modelIn = new
FileInputStream("C:/Users/Lenovo/Documents/en-is-covid-maxent.bin");
                    } catch (FileNotFoundException e) {
                        e.printStackTrace();
                    }
                    DoccatModel model1 = null;
                    try {
                        model1 = new DoccatModel(modelIn);
```

```
                    } catch (IOException e) {
                        e.printStackTrace();
                    }

                    DocumentCategorizer doccat1 = new
DocumentCategorizerME(model1);
                    double[] aProbs1 = doccat1.categorize(anaInput);
                    probc1=aProbs1[0];
                    probc2=aProbs1[1];
                    if(doccat1.getBestCategory(aProbs1).equals("Corona")&&
aProbs1[0]!=0.5) {
                        flag=true;
                        try {
                            modelIn = new
FileInputStream("C:/Users/Lenovo/Documents/en-covid-classifier-
maxent.bin");
                        } catch (FileNotFoundException e) {
                            e.printStackTrace();
                        }
                        DoccatModel model = null;
                        try {
                            model = new DoccatModel(modelIn);
                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                        DocumentCategorizer doccat = new
DocumentCategorizerME(model);
                        double[] aProbs = doccat.categorize(anaInput);
                        probsc1=aProbs[0];
                        probsc2=aProbs[1];
                        probsc3=aProbs[2];
                        probsc4=aProbs[3];
                    }
                    else{
                        flag=false;
                    }

                };

            InputStream modelIn = new
FileInputStream("C:/Users/Lenovo/Documents/en-is-covid-percep.bin");
            DoccatModel model1 = new DoccatModel(modelIn);
            DocumentCategorizer doccat1 = new
DocumentCategorizerME(model1);
```

```
            //Taking input
            String inputString=maintext.getText();
            String tempInput=inputString;
            inputString=inputString.toLowerCase();

            String[] stopwords = {"i", "me", "my", "myself", "we", "our",
"ours", "ourselves", "you", "you're", "you've", "you'll", "you'd", "your",
"yours",
                    "yourself", "yourselves", "he", "him", "his",
"himself", "she", "she's", "her", "hers", "herself", "it", "it's", "its",
"itself",
                    "they", "them", "their", "theirs", "themselves",
"what", "which", "who", "whom", "this", "that", "that'll", "these",
"those", "am",
                    "is", "are", "was", "were", "be", "been", "being",
"have", "has", "had", "having", "do", "does", "did", "doing",
                    "a", "an", "the", "and", "but", "if", "or", "because",
"as", "until", "while", "of", "at", "by", "for",
                    "with", "about", "against", "between", "into",
"through", "during", "before", "after", "above", "below", "to", "from",
"up", "down",
                    "in", "out", "on", "off", "over", "under", "again",
"further", "then", "once", "here", "there", "when", "where", "why",
                    "how", "all", "any", "both", "each", "few", "more",
"most", "other", "some", "such", "no", "nor", "not", "only",
                    "own", "same", "so", "than", "too", "very", "s", "t",
"can", "will", "just", "don", "don't", "should", "should've",
                    "now", "d", "ll", "m", "o", "re", "ve", "y", "ain",
"aren", "aren't", "couldn", "couldn't", "didn", "didn't",
                    "doesn", "doesn't", "hadn", "hadn't", "hasn", "hasn't",
"haven", "haven't", "isn", "isn't", "ma", "mightn", "mightn't", "mustn",
"mustn't",
                    "shan", "shan't", "shouldn", "shouldn't", "wasn",
"wasn't", "weren", "weren't", "won", "won't", "wouldn", "wouldn't"};

            for(int i=0;i< stopwords.length;i++) {
                inputString = inputString.replaceAll(" " + stopwords[i] + "
", " ");
            }
            System.out.println(inputString);
            String[] docWords1 = inputString.split(" ");
```

```
            double[] aProbs1 = doccat1.categorize(docWords1);
            System.out.println(doccat1.getCategory(0) + " : " + aProbs1[0]
+ " : " + doccat1.getCategory(1) + " : " + aProbs1[1] + " : " +
doccat1.getBestCategory(aProbs1) + " : " +
doccat1.getBestCategory(aProbs1).equals("Corona"));

            loadAnalysisChart(analysisPC,aProbs1[0],aProbs1[1]);

            ///////Starting an new Thread/////////
            anaInput=docWords1;
            Thread t=new Thread(analysisPage);
            t.start();
            t.join();
            loadAnalysisChart(analysisMEC,probc1,probc2);

            if(flag==true)
            {
                analysisl3.setVisible(false);
                loadchart(analysisMESC,probsc1,probsc2,probsc3,probsc4);
            }
            else{
                analysisl3.setVisible(true);
                analysisMESC.setVisible(false);
                analysisl3.setText("This will display only when category is
covid.");
            }

            if(doccat1.getBestCategory(aProbs1).equals("Corona")&&
aProbs1[0]!=0.5) {

                //Load the model
                namespane1.setVisible(false);
                datespane1.setVisible(false);
                locationspane1.setVisible(false);
                line1.setVisible(false);
                line2.setVisible(false);
                dates.setVisible(false);
                locations.setVisible(false);
                lineSub.setVisible(true);
                category.setText("COVID");
                analysisl4.setVisible(false);
                modelIn = new
FileInputStream("C:/Users/Lenovo/Documents/en-covid-category-percep.bin");
                DoccatModel model = new DoccatModel(modelIn);
```

```
                DocumentCategorizer doccat = new
DocumentCategorizerME(model);

                // test the model file by subjecting it to prediction
                String[] docWords = inputString.replaceAll("[^A-Za-z]", "
").split(" ");
                double[] aProbs = doccat.categorize(docWords);

                // print the probabilities of the categories
                double c,r,o,s;
                System.out.println("\n--------------------------------
\nCategory : Probability\n---------------------------------");
                for (int i = 0; i < doccat.getNumberOfCategories(); i++) {
                    System.out.println(doccat.getCategory(i) + " : " +
aProbs[i]);
                }
                c=aProbs[0];
                r=aProbs[1];
                o=aProbs[2];
                s=aProbs[3];
                loadchart(pieSub,c,r,o,s);
                loadchart(analysisPSC,c,r,o,s);
                System.out.println("---------------------------------");
                double a = 0, b = 0;
                int p = 0, q = 0;
                for (int i = 0; i < doccat.getNumberOfCategories(); i++) {
                    if (aProbs[i] > b) {
                        b = aProbs[i];
                        q = i;
                    }
                    if (aProbs[i] < b && a < aProbs[i]) {
                        a = aProbs[i];
                        p = i;
                    }
                }
                if (doccat.getBestCategory(aProbs) == "Corona") {
                    System.out.println("\n" +
doccat.getBestCategory(aProbs) + " : is the predicted category for the
given text.");
                    System.out.println("\n" + doccat.getCategory(p) + " :
is the predicted sub-category for the given text.");
                    subCategory.setText(doccat.getCategory(p));
                } else {
                    System.out.println("\n" + "Corona" + " : is the
```

```
predicted category for the given text.");
                    System.out.println("\n" + doccat.getCategory(q) + " :
is the predicted sub-category for the given text.");
                    subCategory.setText(doccat.getCategory(q));
                }
            }
            else
            {

                analysisl4.setVisible(true);
                category.setText("NON-COVID");
                subCategory.setText("NONE");
                lineSub.setVisible(false);
                pieSub.setVisible(false);
                pietext.setVisible(false);
                namespane1.setVisible(true);
                datespane1.setVisible(true);
                locationspane1.setVisible(true);
                line1.setVisible(true);
                line2.setVisible(true);
                dates.setVisible(true);
                locations.setVisible(true);
                names.setVisible(true);
                names.setText("Names");

                analysisPSC.setVisible(false);
                analysisl4.setText("This will display only when category is
covid.");
                nlp(tempInput,namespane1,datespane1, locationspane1);

            }
        } catch (IOException | InterruptedException e) {
            System.out.println("An exception in reading the training file.
Please check.");
            e.printStackTrace();
        }
    }


    private void initialize() {

    }

    public void page1(ActionEvent actionEvent) {
```

```
        pane1.setVisible(true);
        pane2.setVisible(false);
        pane3.setVisible(false);
        pane4.setVisible(false);
        pane5.setVisible(false);
    }

    public void page2(ActionEvent actionEvent) {
        pane1.setVisible(false);
        pane2.setVisible(true);
        pane3.setVisible(false);
        pane4.setVisible(false);
        pane5.setVisible(false);
    }

    public void page3(ActionEvent actionEvent) {
        pane1.setVisible(false);
        pane2.setVisible(false);
        pane3.setVisible(true);
        pane4.setVisible(false);
        pane5.setVisible(false);
    }

    public void page4(ActionEvent actionEvent) {
        pane1.setVisible(false);
        pane2.setVisible(false);
        pane3.setVisible(false);
        pane4.setVisible(true);
        pane5.setVisible(false);
    }
    public void page5(ActionEvent actionEvent) {
        pane1.setVisible(false);
        pane2.setVisible(false);
        pane3.setVisible(false);
        pane4.setVisible(false);
        pane5.setVisible(true);
    }


    public void handlenpl(ActionEvent actionEvent) throws IOException {
        nlp(maintext1.getText(),namespane2,datespane2,locationspane2);
    }
    public void nlp(String tempInput,TextArea namespane,TextArea datespane,
TextArea locationspane) throws IOException {
```

```
        InputStream inputStreamTokenizer = new

FileInputStream("C:/Users/Lenovo/Documents/OpenNLP_models/en-token.bin");
        TokenizerModel tokenModel = new
TokenizerModel(inputStreamTokenizer);
        //Instantiating the TokenizerME class
        TokenizerME tokenizer = new TokenizerME(tokenModel);
        //Tokenizing the sentence in to a string array
        String tokens[] = tokenizer.tokenize(tempInput);
        NlpName namesFind=new
NlpName("C:/Users/Lenovo/Documents/OpenNLP_models/en-ner-
person.bin",tokens);
        String names[]=namesFind.execute(tokens);
        String str="";
        for(String s:names)
        {
            str=str+s+"\n";
        }
        namespane.setText(str);

        ////////////////LOCATION//////////////////
        NlpLocation locationFind=new
NlpLocation("C:/Users/Lenovo/Documents/OpenNLP_models/en-ner-
location.bin",tokens);
        String location[]=locationFind.execute(tokens);
        str="";
        for(String s:location)
        {
            str=str+s+"\n";
        }
        locationspane.setText(str);
        ////////////////////DATE//////////////////
        NlpDate dateFind=new
NlpDate("C:/Users/Lenovo/Documents/OpenNLP_models/en-ner-date.bin",tokens);
        String dates[]=dateFind.execute(tokens);
        str="";
        for(String s:dates)
        {
            str=str+s+"\n";
        }
        datespane.setText(str);
    }
}
```

```
interface Othersnlp{
    public String[] execute(String tokens[]);
}

class OthersNlpTemplate implements  Othersnlp{
    private InputStream inputStreamNameFinder;
    private TokenNameFinderModel model;
    public Span nameSpans[];
    private NameFinderME nameFinder;
    OthersNlpTemplate(String path,String tokens[]) throws IOException {
        inputStreamNameFinder = new FileInputStream(path);
        model = new TokenNameFinderModel(inputStreamNameFinder);
        nameFinder = new NameFinderME(model);
        nameSpans = nameFinder.find(tokens);
    }
    public String[] execute(String tokens[])
    {
        String str[] = new String[0];
        for(Span s: nameSpans)
            System.out.println(s.toString()+"  "+tokens[s.getStart()]);
        return str;
    }
}

class NlpName extends OthersNlpTemplate{
    NlpName(String path,String tokens[]) throws IOException {
        super(path,tokens);
    }
    public String[] execute(String tokens[])
    {
        String str[];
        if(nameSpans.length==0)
            str=new String[1];
        else
            str=new String[nameSpans.length];
        int i=0;
        System.out.println("\nAll the person names present in the given
text are:");
        for(Span s: nameSpans) {
            str[i]=s.toString() + "  " + tokens[s.getStart()];
            i++;
            System.out.println(s.toString() + "  " + tokens[s.getStart()]);
        }
```

```
        if(i==0)
        {
            str[i]="<---None--->";
            System.out.println("<---None--->");
        }
        return str;
    }
}

class NlpLocation extends OthersNlpTemplate{
    NlpLocation(String path,String tokens[]) throws IOException {
        super(path,tokens);
    }
    public String[] execute(String tokens[])
    {
        String str[];
        if(nameSpans.length==0)
            str=new String[1];
        else
            str=new String[nameSpans.length];
        int i=0;
        System.out.println("\nAll the Locations present in the given text
are:");
        for(Span s: nameSpans) {
            str[i]=s.toString() + "  " + tokens[s.getStart()];
            System.out.println(s.toString() + "  " + tokens[s.getStart()]);
            i++;
        }
        if(i==0)
        {
            str[i]="<---None--->";
            System.out.println("<---None--->");
        }
        return str;
    }
}

class NlpDate extends OthersNlpTemplate {
    NlpDate(String path,String tokens[]) throws IOException {
        super(path,tokens);
    }
    public String[] execute(String tokens[])
    {
        String str[];
```

```
        if(nameSpans.length==0)
            str=new String[1];
        else
            str=new String[nameSpans.length];
        int i=0;
        System.out.println("\nAll the Dates present in the given text
are:");
        for(Span s: nameSpans)
        {
            str[i]=s.toString() + "  " + tokens[s.getStart()];
            System.out.println(s.toString()+"  "+tokens[s.getStart()]);
            i++;
        }
        if(i==0)
        {
            str[i]="<---None--->";
            System.out.println("<---None--->");
        }
        return str;
    }
}
```

Model Trainer

```
import java.io.BufferedOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

import opennlp.tools.doccat.DoccatFactory;
import opennlp.tools.doccat.DoccatModel;
import opennlp.tools.doccat.DocumentCategorizerME;
import opennlp.tools.doccat.DocumentSampleStream;
import opennlp.tools.util.InputStreamFactory;
import opennlp.tools.util.MarkableFileInputStreamFactory;
import opennlp.tools.util.ObjectStream;
import opennlp.tools.util.PlainTextByLineStream;
import opennlp.tools.util.TrainingParameters;

public class ClassifierTrainer {
    static DoccatModel model;
    public static void main(String[] args) {
        try {
            // read the training data
```

```
            InputStreamFactory dataIn = new
MarkableFileInputStreamFactory(new File("C:/Users/Lenovo/Documents/en-
covid-category.train"));
            ObjectStream lineStream = new PlainTextByLineStream(dataIn,
"UTF-8");
            ObjectStream sampleStream = new
DocumentSampleStream(lineStream);
            // define the training parameters
            TrainingParameters params = new TrainingParameters();
            params.put(TrainingParameters.ITERATIONS_PARAM, 10+"");
            params.put(TrainingParameters.CUTOFF_PARAM, 0+"");

            // create a model from training data
            model = DocumentCategorizerME.train("en", sampleStream, params,
new DoccatFactory());
            System.out.println("\nModel is successfully trained.");

            // save the model to local
            BufferedOutputStream modelOut = new BufferedOutputStream(new
FileOutputStream("C:/Users/Lenovo/Documents/en-covid-classifier-
maxent.bin"));
            model.serialize(modelOut);
            System.out.println("\nTrained Model is saved locally at :
C:/Users/Lenovo/Documents/en-covid-classifier-maxent.bin");
            System.out.println(model);
        }
            catch (IOException e) {
                System.out.println("An exception in reading the training
file. Please check.");
                e.printStackTrace();
        }
    }
}
```

dash.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.chart.CategoryAxis?>
<?import javafx.scene.chart.LineChart?>
<?import javafx.scene.chart.NumberAxis?>
<?import javafx.scene.chart.PieChart?>
<?import javafx.scene.control.Button?>
```

```
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextArea?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.shape.Circle?>
<?import javafx.scene.shape.Line?>
<?import javafx.scene.text.Font?>
<?import javafx.scene.text.Text?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-
Infinity" minWidth="-Infinity" prefHeight="530.0" prefWidth="957.0"
style="-fx-border-radius: 20; -fx-background-radius: 20;"
xmlns="http://javafx.com/javafx/16" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="main.java.sample.controllers.DashController">
   <children>
      <Pane prefHeight="530.0" prefWidth="965.0" style="-fx-background-
color: #FFF; -fx-background-radius: 20;">
         <children>
            <VBox prefHeight="530.0" prefWidth="207.0" style="-fx-
background-color: #E5E5E5;">
               <children>
                  <ImageView fitHeight="144.0" fitWidth="221.0"
pickOnBounds="true" preserveRatio="true">
                     <image>
                        <Image url="@../../imgs/logonew.jpg" />
                     </image>
                  </ImageView>
                  <Button alignment="BASELINE_LEFT" graphicTextGap="10.0"
mnemonicParsing="false" onAction="#page1" prefHeight="37.0"
prefWidth="210.0" style="-fx-cursor: hand;"
stylesheets="@../utils/fullpackstyling.css" text="Covid Classifer">
                     <padding>
                        <Insets left="40.0" />
                     </padding>
                     <graphic>
                        <ImageView fitHeight="20.0" fitWidth="26.0"
pickOnBounds="true" preserveRatio="true">
                           <image>
                              <Image
url="@../../imgs/icons8_details_24px_3.png" />
```

```
                           </image>
                        </ImageView>
                     </graphic>
                  </Button>
                  <Button alignment="BASELINE_LEFT" graphicTextGap="10.0"
layoutX="10.0" layoutY="10.0" mnemonicParsing="false" onAction="#page2"
prefHeight="37.0" prefWidth="210.0" style="-fx-cursor: hand;"
stylesheets="@../utils/fullpackstyling.css" text="Extract Info">
                     <VBox.margin>
                        <Insets />
                     </VBox.margin>
                     <padding>
                        <Insets left="40.0" />
                     </padding>
                     <graphic>
                        <ImageView fitHeight="20.0" fitWidth="26.0"
pickOnBounds="true" preserveRatio="true">
                           <image>
                              <Image
url="@../../imgs/icons8_compass_24px_1.png" />
                           </image>
                        </ImageView>
                     </graphic>
                  </Button>
                  <Button alignment="BASELINE_LEFT" graphicTextGap="10.0"
layoutX="10.0" layoutY="173.0" mnemonicParsing="false" onAction="#page5"
prefHeight="37.0" prefWidth="210.0" style="-fx-cursor: hand;"
stylesheets="@../utils/fullpackstyling.css" text="Comparison">
                     <padding>
                        <Insets left="40.0" />
                     </padding>
                     <graphic>
                        <ImageView fitHeight="20.0" fitWidth="26.0"
pickOnBounds="true" preserveRatio="true">
                           <image>
                              <Image url="@../../imgs/icons8-increase-
profits-96.png" />
                           </image>
                        </ImageView>
                     </graphic>
                  </Button>
                  <Button alignment="BASELINE_LEFT" graphicTextGap="10.0"
layoutX="10.0" layoutY="47.0" mnemonicParsing="false" onAction="#page3"
prefHeight="37.0" prefWidth="210.0" style="-fx-cursor: hand;"
```

```
stylesheets="@../utils/fullpackstyling.css" text="How it works?">
                     <VBox.margin>
                        <Insets />
                     </VBox.margin>
                     <padding>
                        <Insets left="40.0" />
                     </padding>
                     <graphic>
                        <ImageView fitHeight="20.0" fitWidth="26.0"
pickOnBounds="true" preserveRatio="true">
                           <image>
                              <Image
url="@../../imgs/icons8_settings_32px.png" />
                           </image>
                        </ImageView>
                     </graphic>
                  </Button>
                  <Button alignment="BASELINE_LEFT" graphicTextGap="10.0"
layoutX="10.0" layoutY="84.0" mnemonicParsing="false" onAction="#page4"
prefHeight="37.0" prefWidth="210.0" style="-fx-cursor: hand;"
stylesheets="@../utils/fullpackstyling.css" text="About us">
                     <VBox.margin>
                        <Insets />
                     </VBox.margin>
                     <padding>
                        <Insets left="40.0" />
                     </padding>
                     <graphic>
                        <ImageView fitHeight="20.0" fitWidth="26.0"
pickOnBounds="true" preserveRatio="true">
                           <image>
                              <Image url="@../../imgs/about-us.png" />
                           </image>
                        </ImageView>
                     </graphic>
                  </Button>
                  <Pane prefHeight="254.0" prefWidth="221.0" style="-fx-
background-radius: 20; -fx-border-radius: 20;">
                     <children>
                        <Circle fx:id="shpActive" fill="WHITE"
layoutX="53.0" layoutY="87.0" radius="7.0" stroke="#1cd95c"
strokeType="INSIDE" />
                        <Circle fill="WHITE" layoutX="53.0" layoutY="120.0"
radius="7.0" stroke="#d9c81c" strokeType="INSIDE" />
```

```
                        <Circle fill="WHITE" layoutX="53.0" layoutY="149.0"
radius="7.0" stroke="#da1c1c" strokeType="INSIDE" />
                        <Line endX="-4.0" endY="-8.0" layoutX="34.0"
layoutY="43.0" startX="148.0" startY="-8.0" stroke="#ddd1d1" />
                        <Label fx:id="lblActive" layoutX="76.0"
layoutY="79.0" text="10 Active" textFill="#4a4747">
                           <font>
                              <Font size="11.0" />
                           </font>
                        </Label>
                        <Label fx:id="lblOffline" layoutX="78.0"
layoutY="112.0" text="2 Offline" textFill="#4a4747">
                           <font>
                              <Font size="11.0" />
                           </font>
                        </Label>
                        <Label fx:id="lblExp" layoutX="77.0"
layoutY="141.0" text="8 Expired" textFill="#4a4747">
                           <font>
                              <Font size="11.0" />
                           </font>
                        </Label>
                        <Label layoutX="36.0" layoutY="43.0"
prefHeight="28.0" prefWidth="210.0" text="India Covid Stats"
underline="true">
                           <font>
                              <Font name="Consolas" size="14.0" />
                           </font>
                        </Label>
                     </children>
                  </Pane>
               </children>
            </VBox>
            <Pane fx:id="pane1" layoutX="203.0" prefHeight="530.0"
prefWidth="754.0" style="-fx-background-color: #fff; -fx-background-radius:
20;">
               <children>
                  <Line endX="598.0" endY="0.5" layoutX="133.0"
layoutY="302.0" startX="-98.5" startY="0.5" stroke="#ddd1d1" />
                  <Line fx:id="line1" endX="3.0" endY="-8.0"
layoutX="271.0" layoutY="312.0" startX="3.0" startY="147.29290771484375"
stroke="#ddd1d1" visible="false" />
                  <Label fx:id="names" layoutX="34.0" layoutY="308.0"
text="Names" textFill="#383839" visible="false">
```

```
                     <font>
                        <Font name="System Bold" size="16.0" />
                     </font>
                  </Label>
                  <Label fx:id="dates" layoutX="288.0" layoutY="308.0"
text="Dates" textFill="#383839" visible="false">
                     <font>
                        <Font name="System Bold" size="16.0" />
                     </font>
                  </Label>
                  <Label fx:id="locations" layoutX="548.0" layoutY="308.0"
text="Locations" textFill="#383839" visible="false">
                     <font>
                        <Font name="System Bold" size="16.0" />
                     </font>
                  </Label>
                  <Line fx:id="line2" endX="3.0" endY="-8.0"
layoutX="518.0" layoutY="312.0" startX="3.0" startY="147.29290771484375"
stroke="#ddd1d1" visible="false" />
                  <Label layoutX="33.0" layoutY="48.0" text="This
application allows you to check whether the given text is related to COVID-
19 or not." textFill="#383839" underline="true">
                     <font>
                        <Font name="System Bold" size="16.0" />
                     </font>
                  </Label>
                  <Circle fill="#f94344" layoutX="732.0" layoutY="22.0"
onMouseClicked="#close" radius="14.0" stroke="BLACK" strokeType="INSIDE" />
                  <Label fx:id="lblClose" alignment="CENTER"
layoutX="718.0" layoutY="14.0" onMouseClicked="#close" prefHeight="17.0"
prefWidth="28.0" style="-fx-cursor: hand;" text="X" textFill="WHITE">
                     <font>
                        <Font name="System Bold" size="12.0" />
                     </font>
                  </Label>
                  <Circle fill="#29ff1f" layoutX="689.0" layoutY="22.0"
onMouseClicked="#min" radius="14.0" stroke="#070707" strokeType="INSIDE" />
                  <Label layoutX="683.0" layoutY="-13.0"
onMouseClicked="#min" style="-fx-cursor: hand;" text="_">
                     <font>
                        <Font size="29.0" />
                     </font>
                  </Label>
                  <TextField fx:id="maintext" layoutX="34.0" layoutY="90.0"
```

```
onAction="#handle" prefHeight="37.0" prefWidth="414.0" promptText="Enter
Text to be classified" style="-fx-cursor: text;" />
                  <Button layoutX="34.0" layoutY="143.0"
mnemonicParsing="false" onAction="#handle" prefHeight="28.0"
prefWidth="100.0" style="-fx-cursor: hand;" text="Classify">
                     <font>
                        <Font name="Consolas" size="12.0" />
                     </font></Button>
                  <TextArea fx:id="category" editable="false"
layoutX="41.0" layoutY="193.0" minHeight="72.0" minWidth="200.0"
nodeOrientation="LEFT_TO_RIGHT" prefHeight="72.0" prefWidth="272.0"
promptText="Category will appear here">
                     <font>
                        <Font name="Consolas Bold" size="41.0" />
                     </font>
                  </TextArea>
                  <TextArea fx:id="subCategory" editable="false"
layoutX="320.0" layoutY="204.0" prefHeight="30.0" prefWidth="180.0"
promptText="Sub-Category">
                     <font>
                        <Font name="Consolas" size="20.0" />
                     </font></TextArea>
                  <Text layoutX="370.0" layoutY="276.0"
strokeType="OUTSIDE" strokeWidth="0.0" text="Sub-Category" />
                  <Text layoutX="150.0" layoutY="283.0"
strokeType="OUTSIDE" strokeWidth="0.0" text="Category" />
                  <PieChart fx:id="pieSub" layoutX="469.0" layoutY="71.0"
prefHeight="216.0" prefWidth="293.0" title="Probabilities" />
                  <Text fx:id="pietext" layoutX="548.0" layoutY="183.0"
strokeType="OUTSIDE" strokeWidth="0.0" text="Pie Chart will display here"
/>
                  <TextArea fx:id="namespane1" editable="false"
layoutX="34.0" layoutY="333.0" prefHeight="181.0" prefWidth="200.0"
visible="false">
                     <font>
                        <Font name="Consolas" size="12.0" />
                     </font></TextArea>
                  <TextArea fx:id="datespane1" editable="false"
layoutX="288.0" layoutY="332.0" prefHeight="181.0" prefWidth="200.0"
visible="false">
                     <font>
                        <Font name="Consolas" size="12.0" />
                     </font></TextArea>
                  <TextArea fx:id="locationspane1" editable="false"
```

```
layoutX="548.0" layoutY="333.0" prefHeight="181.0" prefWidth="187.0"
visible="false">
                     <font>
                        <Font name="Consolas" size="12.0" />
                     </font></TextArea>
                  <LineChart fx:id="lineSub" layoutX="8.0" layoutY="302.0"
prefHeight="222.0" prefWidth="723.0">
                    <xAxis>
                      <CategoryAxis side="BOTTOM" />
                    </xAxis>
                    <yAxis>
                      <NumberAxis side="LEFT" />
                    </yAxis>
                  </LineChart>
               </children>
            </Pane>
            <Pane fx:id="pane2" layoutX="203.0" prefHeight="530.0"
prefWidth="778.0" style="-fx-background-color: #fff; -fx-background-radius:
20;" visible="false">
               <children>
                  <Line endX="598.0" endY="0.5" layoutX="134.0"
layoutY="264.0" startX="-98.5" startY="0.5" stroke="#ddd1d1" />
                  <Line endX="3.0" endY="-8.0" layoutX="270.0"
layoutY="274.0" startX="3.0" startY="147.29290771484375" stroke="#ddd1d1"
/>
                  <Label layoutX="33.0" layoutY="275.0" text="Names"
textFill="#383839">
                     <font>
                        <Font name="System Bold" size="16.0" />
                     </font>
                  </Label>
                  <Label layoutX="288.0" layoutY="275.0" text="Dates"
textFill="#383839">
                     <font>
                        <Font name="System Bold" size="16.0" />
                     </font>
                  </Label>
                  <Label layoutX="545.0" layoutY="275.0" text="Locations"
textFill="#383839">
                     <font>
                        <Font name="System Bold" size="16.0" />
                     </font>
                  </Label>
                  <Line endX="3.0" endY="-8.0" layoutX="518.0"
```

```
layoutY="274.0" startX="3.0" startY="147.29290771484375" stroke="#ddd1d1"
/>
                  <Label layoutX="33.0" layoutY="48.0" text="You can
extract useful information from the text like Names, Dates, Locatons"
textFill="#383839" underline="true">
                     <font>
                        <Font name="System Bold" size="16.0" />
                     </font>
                  </Label>
                  <Circle fill="#f94344" layoutX="732.0" layoutY="22.0"
radius="14.0" stroke="BLACK" strokeType="INSIDE" />
                  <Circle fill="WHITE" layoutX="639.0" layoutY="425.0"
radius="10.0" stroke="WHITE" strokeType="INSIDE" />
                  <Label fx:id="lblClose1" alignment="CENTER"
layoutX="718.0" layoutY="14.0" onMouseClicked="#close" prefHeight="17.0"
prefWidth="28.0" style="-fx-cursor: hand;" text="X" textFill="WHITE">
                     <font>
                        <Font name="System Bold" size="12.0" />
                     </font>
                  </Label>
                  <Circle fill="#29ff1f" layoutX="689.0" layoutY="22.0"
radius="14.0" stroke="#070707" strokeType="INSIDE" />
                  <Label layoutX="683.0" layoutY="-13.0"
onMouseClicked="#min" style="-fx-cursor: hand;" text="_">
                     <font>
                        <Font size="29.0" />
                     </font>
                  </Label>
                  <TextField fx:id="maintext1" layoutX="34.0"
layoutY="90.0" onAction="#handlenpl" prefHeight="37.0" prefWidth="414.0"
promptText="Enter Text to extract information" style="-fx-cursor: text;" />
                  <Button layoutX="34.0" layoutY="143.0"
mnemonicParsing="false" onAction="#handlenpl" prefHeight="28.0"
prefWidth="100.0" style="-fx-cursor: hand;" text="Extract">
                     <font>
                        <Font name="Consolas" size="12.0" />
                     </font>
                  </Button>
                  <TextArea fx:id="namespane2" editable="false"
layoutX="34.0" layoutY="300.0" prefHeight="200.0" prefWidth="200.0"
promptText="Names will appear here">
                     <font>
                        <Font name="Consolas" size="12.0" />
                     </font></TextArea>
```

```
                  <TextArea fx:id="datespane2" editable="false"
layoutX="285.0" layoutY="300.0" prefHeight="200.0" prefWidth="200.0"
promptText="Dates will appear here">
                     <font>
                        <Font name="Consolas" size="12.0" />
                     </font></TextArea>
                  <TextArea fx:id="locationspane2" editable="false"
layoutX="539.0" layoutY="300.0" prefHeight="200.0" prefWidth="200.0"
promptText="Location will appeat here">
                     <font>
                        <Font name="Consolas" size="12.0" />
                     </font></TextArea>
               </children>
            </Pane>
            <Pane fx:id="pane3" layoutX="203.0" prefHeight="530.0"
prefWidth="788.0" style="-fx-background-color: #fff; -fx-background-radius:
20;" visible="false">
               <children>
                  <Circle fill="#f94344" layoutX="732.0" layoutY="22.0"
radius="14.0" stroke="BLACK" strokeType="INSIDE" />
                  <Circle fill="WHITE" layoutX="639.0" layoutY="425.0"
radius="10.0" stroke="WHITE" strokeType="INSIDE" />
                  <Label fx:id="lblClose11" alignment="CENTER"
layoutX="718.0" layoutY="14.0" onMouseClicked="#close" prefHeight="17.0"
prefWidth="28.0" style="-fx-cursor: hand;" text="X" textFill="WHITE">
                     <font>
                        <Font name="System Bold" size="12.0" />
                     </font>
                  </Label>
                  <Circle fill="#29ff1f" layoutX="689.0" layoutY="22.0"
radius="14.0" stroke="#070707" strokeType="INSIDE" />
                  <Label layoutX="683.0" layoutY="-13.0"
onMouseClicked="#min" style="-fx-cursor: hand;" text="_">
                     <font>
                        <Font size="29.0" />
                     </font>
                  </Label>
                  <TextArea fx:id="howTo" editable="false" layoutY="46.0"
prefHeight="484.0" prefWidth="754.0" text="                        How this
application works?&#10;&#10;1. Covid Classifier: Available on the left pane
is used to classify input text as covid or non-covid. If it falls in covid
category, it is further classified as either of general Corona, Request,
Offer or Statistics. Graphical representation of probabilities of sub -
categories is also displayed. If the text falls in non-covid category,
```

```
important information like Location, Names and Dates from the inputted text are displayed.&#10;2.  Extract Info: Extracts and prints important information like Location, Names and Dates from the inputted text.&#10;3. How it works?: It is a guide on how this application works.&#10;4. About us: This section provides information about the developers of this application.&#10;5. The bottom part of the left pane contains real-time Covid related data of India.&#10;6. The application has Minimise and Close button on top right side of the screen." wrapText="true">
                     <font>
                        <Font name="Consolas" size="16.0" />
                     </font>
                  </TextArea>
               </children>
            </Pane>
            <Pane fx:id="pane4" layoutX="205.0" prefHeight="530.0" prefWidth="810.0" style="-fx-background-color: #fff; -fx-background-radius: 20;" visible="false">
               <children>
                  <Circle fill="#f94344" layoutX="732.0" layoutY="22.0" radius="14.0" stroke="BLACK" strokeType="INSIDE" />
                  <Circle fill="WHITE" layoutX="639.0" layoutY="425.0" radius="10.0" stroke="WHITE" strokeType="INSIDE" />
                  <Label fx:id="lblClose111" alignment="CENTER" layoutX="718.0" layoutY="14.0" onMouseClicked="#close" prefHeight="17.0" prefWidth="28.0" style="-fx-cursor: hand;" text="X" textFill="WHITE">
                     <font>
                        <Font name="System Bold" size="12.0" />
                     </font>
                  </Label>
                  <Circle fill="#29ff1f" layoutX="689.0" layoutY="22.0" radius="14.0" stroke="#070707" strokeType="INSIDE" />
                  <Label layoutX="683.0" layoutY="-13.0" onMouseClicked="#min" style="-fx-cursor: hand;" text="_">
                     <font>
                        <Font size="29.0" />
                     </font>
                  </Label>
                  <TextField fx:id="aboutUs" alignment="CENTER" editable="false" layoutX="236.0" layoutY="66.0" text="About Us">
                     <font>
                        <Font name="System Bold" size="20.0" />
                     </font>
                  </TextField>
                  <TextArea fx:id="aboutUsDetails" editable="false"
```

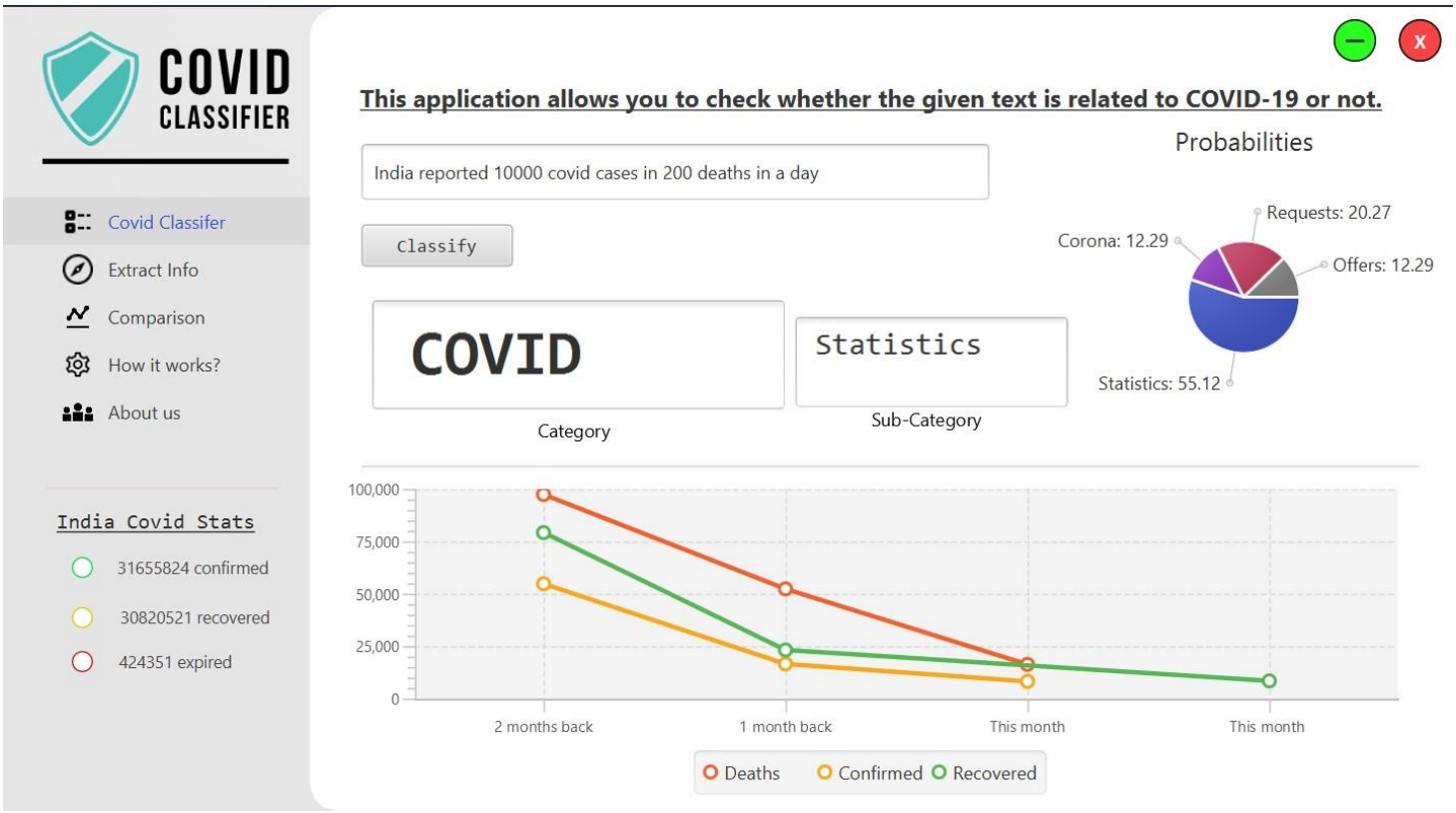
```
layoutX="37.0" layoutY="134.0" prefHeight="335.0" prefWidth="664.0"
text="This project is developed by below mentioned 4th semester Information
Science and Engineering students of R V College of Engineering as a part of
Object Oriented Programming in Java course.&#10;1. Sahil Sharma -
1RV19IS046&#10;2. Shivam Prajapati - 1RV19IS049" wrapText="true">
                     <font>
                        <Font name="Consolas" size="16.0" />
                     </font>
                  </TextArea>
               </children>
            </Pane>
            <Pane fx:id="pane5" layoutX="203.0" prefHeight="530.0"
prefWidth="810.0" style="-fx-background-color: #fff; -fx-background-radius:
20;" visible="false">
               <children>
                  <Circle fill="#f94344" layoutX="732.0" layoutY="22.0"
radius="14.0" stroke="BLACK" strokeType="INSIDE" />
                  <Circle fill="WHITE" layoutX="639.0" layoutY="425.0"
radius="10.0" stroke="WHITE" strokeType="INSIDE" />
                  <Label fx:id="lblClose1111" alignment="CENTER"
layoutX="718.0" layoutY="14.0" onMouseClicked="#close" prefHeight="17.0"
prefWidth="28.0" style="-fx-cursor: hand;" text="X" textFill="WHITE">
                     <font>
                        <Font name="System Bold" size="12.0" />
                     </font>
                  </Label>
                  <Circle fill="#29ff1f" layoutX="689.0" layoutY="22.0"
radius="14.0" stroke="#070707" strokeType="INSIDE" />
                  <Label layoutX="683.0" layoutY="-13.0"
onMouseClicked="#min" style="-fx-cursor: hand;" text="_">
                     <font>
                        <Font size="29.0" />
                     </font>
                  </Label>
                  <Label layoutX="27.0" layoutY="36.0" text="-&gt;
Comparison of ML model using different algorithms" underline="true">
                     <font>
                        <Font name="Consolas" size="14.0" />
                     </font>
                  </Label>
                  <Line endX="-100.0" endY="382.0" layoutX="490.0"
layoutY="147.0" startX="-100.0" startY="-76.33333587646484"
stroke="#ddd1d1" />
                  <Line endX="552.3333740234375" endY="-7.62939453125E-6"
```

```
layoutX="216.0" layoutY="114.0" startX="-181.6666717529297" startY="-
7.62939453125E-6" stroke="#161616" />
                  <Label layoutX="155.0" layoutY="83.0" text="Max-Entropy"
underline="true">
                     <font>
                        <Font name="Consolas Bold" size="20.0" />
                     </font>
                  </Label>
                  <Label layoutX="529.0" layoutY="83.0" text="Perceptron"
underline="true">
                     <font>
                        <Font name="Consolas Bold" size="20.0" />
                     </font>
                  </Label>
                  <PieChart fx:id="analysisMEC" animated="false"
cache="true" cacheHint="QUALITY" cacheShape="false" centerShape="false"
clockwise="false" layoutX="81.0" layoutY="111.0" legendVisible="false"
prefHeight="200.0" prefWidth="265.0" scaleShape="false" title="Category" />
                  <PieChart fx:id="analysisPC" animated="false"
cache="true" cacheHint="QUALITY" cacheShape="false" centerShape="false"
clockwise="false" layoutX="456.0" layoutY="111.0" legendVisible="false"
prefHeight="200.0" prefWidth="272.0" scaleShape="false" title="Category" />
                  <PieChart fx:id="analysisMESC" animated="false"
cache="true" cacheHint="QUALITY" cacheShape="false" centerShape="false"
clockwise="false" labelLineLength="15.0" layoutX="76.0" layoutY="311.0"
legendVisible="false" prefHeight="230.0" prefWidth="259.0"
scaleShape="false" title="Sub-Category" />
                  <PieChart fx:id="analysisPSC" animated="false"
cache="true" cacheHint="QUALITY" cacheShape="false" centerShape="false"
clockwise="false" layoutX="456.0" layoutY="314.0" legendVisible="false"
prefHeight="222.0" prefWidth="281.0" scaleShape="false" title="Sub-
Category" />
                  <Label fx:id="analysisl1" layoutX="29.0" layoutY="214.0"
text="Pie Chart will display here once an input is given for covid
classifier" />
                  <Label fx:id="analysisl3" layoutX="108.0" layoutY="426.0"
text="Pie Chart will display here" />
                  <Label fx:id="analysisl2" layoutX="529.0" layoutY="214.0"
text="Pie Chart will display here" />
                  <Label fx:id="analysisl4" layoutX="503.0" layoutY="426.0"
text="Pie Chart will display here" />
                  <Line endX="383.66668701171875" layoutX="389.0"
layoutY="311.0" startX="-353.0" stroke="#ddd1d1" />
               </children>
```

```
            </Pane>

        </children>
    </Pane>
  </children>
</AnchorPane>
```

OUTPUTS



COVID CLASSIFIER

- Covid Classifer
- Extract Info
- Comparison
- How it works?
- About us

India Covid Stats

- 31655824 confirmed
- 30820521 recovered
- 424351 expired

This application allows you to check whether the given text is related to COVID-19 or not.

John and Peter lives in London and were born in 2002

Classify

NON-COVID

Category

NONE

Sub-Category

Names

```
[0..1) person  John
[2..3) person  Peter
```

Dates

```
[10..11) date  2002
```

Locations

```
[5..6) location  London
```

COVID CLASSIFIER

- Covid Classifer
- Extract Info
- Comparison
- How it works?
- About us

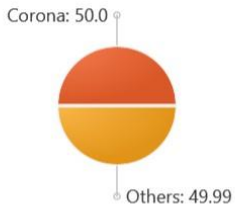
India Covid Stats

- 31655824 confirmed
- 30820521 recovered
- 424351 expired

-> Comparison of ML model using different algorithms

Max-Entropy

Category

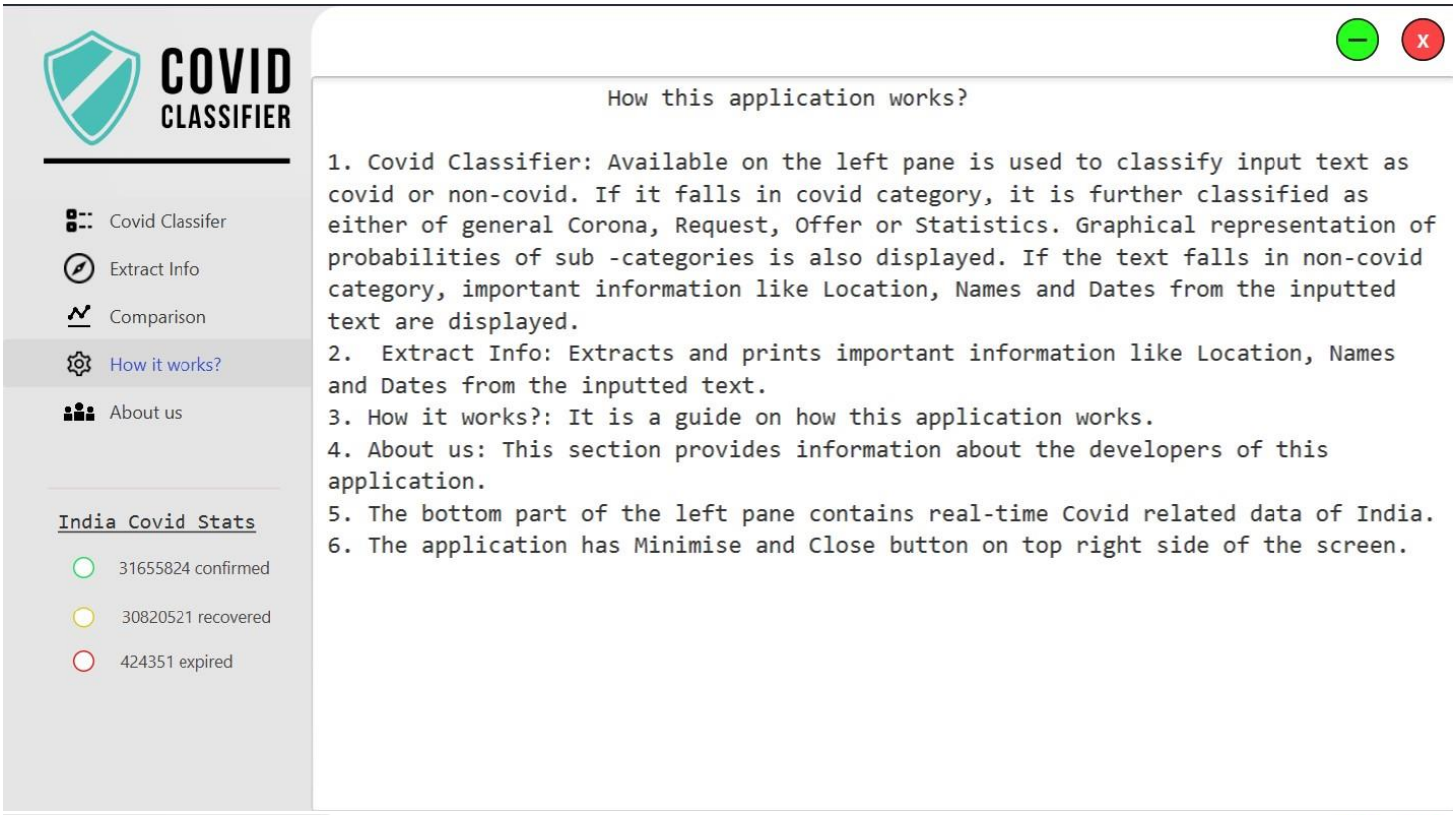


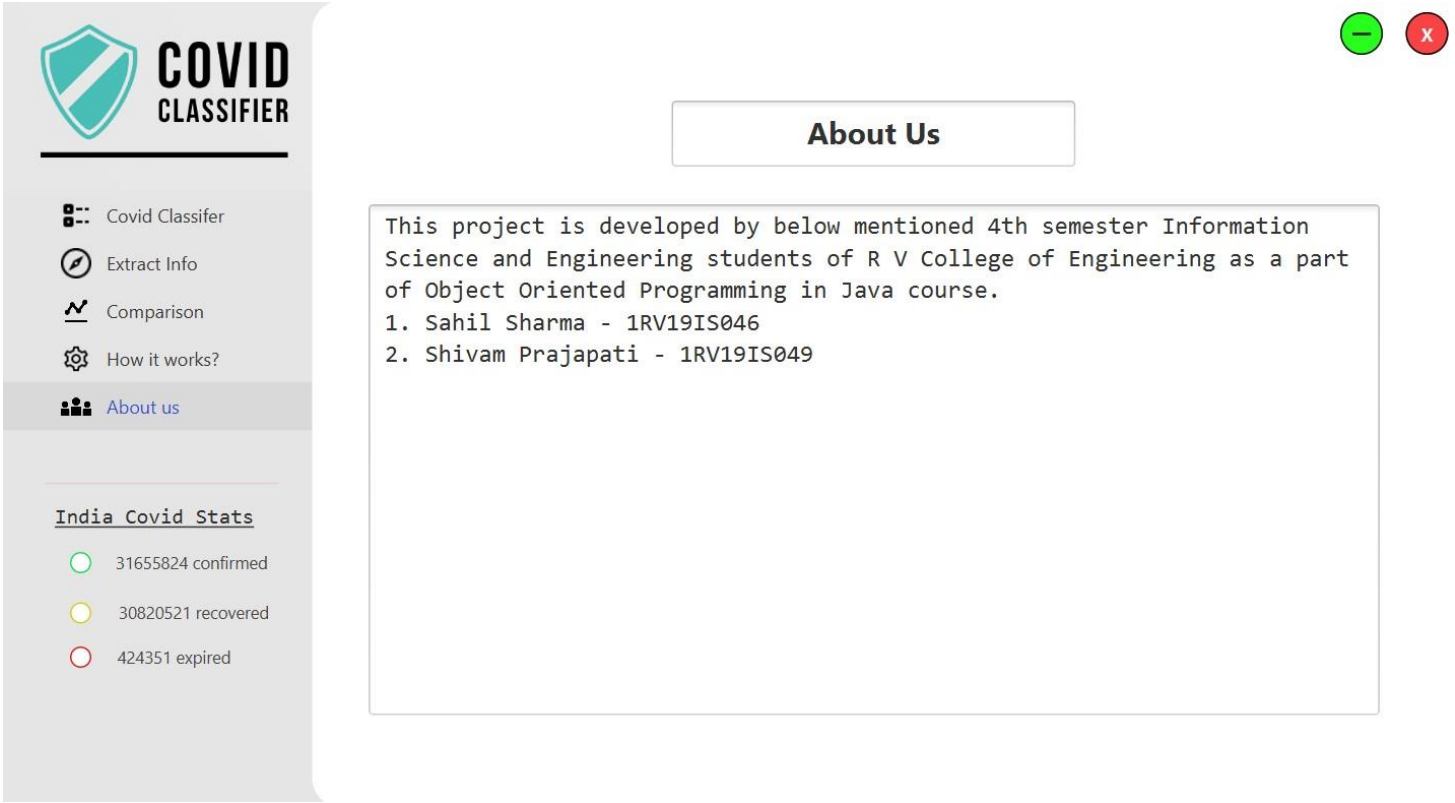
Sub-Category

Perceptron

Category

Sub-Category


COVID
CLASSIFIER
Covid Classifer
Extract Info
Comparison
How it works?
About us
India Covid Stats
31655824 confirmed
30820521 recovered
424351 expired
How this application works?
1. Covid Classifier: Available on the left pane is used to classify input text as covid or non-covid. If it falls in covid category, it is further classified as either of general Corona, Request, Offer or Statistics. Graphical representation of probabilities of sub -categories is also displayed. If the text falls in non-covid category, important information like Location, Names and Dates from the inputted text are displayed.
2. Extract Info: Extracts and prints important information like Location, Names and Dates from the inputted text.
3. How it works?: It is a guide on how this application works.
4. About us: This section provides information about the developers of this application.
5. The bottom part of the left pane contains real-time Covid related data of India.
6. The application has Minimise and Close button on top right side of the screen.


COVID
CLASSIFIER
Covid Classifer
Extract Info
Comparison
How it works?
About us
India Covid Stats
31655824 confirmed
30820521 recovered
424351 expired
About Us
This project is developed by below mentioned 4th semester Information Science and Engineering students of R V College of Engineering as a part of Object Oriented Programming in Java course.
1. Sahil Sharma - 1RV19IS046
2. Shivam Prajapati - 1RV19IS049