As we use MVC in our project so there are many pages that we implement the code in according to our MODELS:

1- He will buy:

```
   //on page-load
        public ActionResult He_will_buy()
        {
            //declaring and sending values from controller to view
            int[] Children = new int[] { 0, 1, 2, 3, 4, 5 };
            ViewBag.Gender = new SelectList(db.Customers.Select(x =>
x.Gender).Distinct());
            ViewBag.Marital_Status = new SelectList(db.Customers.Select(x =>
x.Marital_Status).Distinct());
            ViewBag.Education = new SelectList(db.Customers.Select(x =>
x.Education).Distinct());
            //for example user can't choose any other occupation
            ViewBag.Occupation = new SelectList(db.Customers.Select(x =>
x.Occupation).Distinct());
            ViewBag.Home_Owner = new SelectList(db.Customers.Select(x =>
x.Home_Owner).Distinct());
            ViewBag.Cars = new SelectList(db.Customers.Select(x => x.Cars).Distinct());
            ViewBag.Commute_Distance = new SelectList(db.Customers.Select(x =>
x.Commute_Distance).Distinct());
            ViewBag.Region = new SelectList(db.Customers.Select(x =>
x.Region).Distinct());
            ViewBag.Children = new SelectList(Children);

            return View();
        }
        //onclick
        [HttpPost, ActionName("He_will_buy")]
        public ActionResult He_will_buyPost()
        {
            //same varibles should be readed by the buttoon and post it to the DB
            int[] Children = new int[] { 0, 1, 2, 3, 4, 5 };
            ViewBag.Gender = new SelectList(db.Customers.Select(x =>
x.Gender).Distinct());
            ViewBag.Marital_Status = new SelectList(db.Customers.Select(x =>
x.Marital_Status).Distinct());
            ViewBag.Education = new SelectList(db.Customers.Select(x =>
x.Education).Distinct());
            ViewBag.Occupation = new SelectList(db.Customers.Select(x =>
x.Occupation).Distinct());
            ViewBag.Home_Owner = new SelectList(db.Customers.Select(x =>
x.Home_Owner).Distinct());
            ViewBag.Cars = new SelectList(db.Customers.Select(x => x.Cars).Distinct());
            ViewBag.Commute_Distance = new SelectList(db.Customers.Select(x =>
x.Commute_Distance).Distinct());
            ViewBag.Region = new SelectList(db.Customers.Select(x =>
x.Region).Distinct());
            ViewBag.Children = new SelectList(Children);

            //requesting the posted value by the user
            string Commute_Distance = Request["Commute_Distance"].ToString();
```

```
            string Education = Request.Form["Education"].ToString();
            string Gender = Request.Form["Gender"].ToString();
            string Home_Owner = Request.Form["Home_Owner"].ToString();
            string Marital_Status = Request.Form["Marital_Status"].ToString();
            string Occupation = Request.Form["Occupation"].ToString();
            string Region = Request.Form["Region"].ToString();
            int getage = Convert.ToInt32(Request.Form["Age"]);
            int getcar = Convert.ToInt32(Request.Form["Cars"]);
            int getchild = Convert.ToInt32(Request.Form["Children"]);
            int getincome = Convert.ToInt32(Request.Form["Income"]);

            //declaring the Admomd connection
            AdomdConnection con = new AdomdConnection("Data Source=WS\\SQLEXPRESS;
Catalog=ADMDataBase");
            con.Open();
            AdomdCommand cmd = new AdomdCommand();
            cmd.Connection = con;
            //setting the query
            string s = @"Select flattened predicthistogram (Buy) from BIKECOBMM natural
prediction join
                                        (select '" + getage + @"' as [Age],
                                         '" + getcar + @"' as [Cars],
                                         '" + getchild + @" ' as [Children],
                                         '" + Commute_Distance + @"' as [Commute
Distance],
                                         '" + Education + @"' as [Education],
                                         '" + Gender + @"' as [Gender],
                                         '" + Home_Owner + @"' as [Home Owner],
                                         '" + getincome + @"' as [Income],
                                         '" + Marital_Status + @"' as [Marital
Status],
                                         '" + Occupation + @"' as [Occupation],
                                         '" + Region + @"' as [Region]) as t";
            cmd.CommandText = s;
            AdomdDataReader dr = cmd.ExecuteReader();
            string final_result = "";

            //bring me the result of string "yes/no" and the int percentage
            if (dr.Read())
            {
                if (dr[0] != null)
                {
                    final_result += dr[0].ToString() + " " + dr[2].ToString();
                    ViewBag.final_resultstring = dr[0].ToString();
                    //send the results to view
                    ViewBag.final_resultint = dr[2].ToString();
                }
                dr.Close();
                con.Close();
            }
            return View();
        }
```

2- Classify Customer:

```
//on page-load
        public ActionResult Classify_customer()
        {
            //declaring values and send it to view
            int[] Children = new int[] { 0, 1, 2, 3, 4, 5 };
            ViewBag.Gender = new SelectList(db.Customers.Select(x =>
x.Gender).Distinct());
            ViewBag.Marital_Status = new SelectList(db.Customers.Select(x =>
x.Marital_Status).Distinct());
            ViewBag.Education = new SelectList(db.Customers.Select(x =>
x.Education).Distinct());
            ViewBag.Occupation = new SelectList(db.Customers.Select(x =>
x.Occupation).Distinct());
            ViewBag.Home_Owner = new SelectList(db.Customers.Select(x =>
x.Home_Owner).Distinct());
            ViewBag.Cars = new SelectList(db.Customers.Select(x => x.Cars).Distinct());
            ViewBag.Commute_Distance = new SelectList(db.Customers.Select(x =>
x.Commute_Distance).Distinct());
            ViewBag.Region = new SelectList(db.Customers.Select(x =>
x.Region).Distinct());
            ViewBag.Children = new SelectList(Children);

            return View();
        }
        //on post
        [HttpPost, ActionName("Classify_customer")]
        public ActionResult Classify_customerPost()
        {
            //setting the values that user should choose from
            int[] Children = new int[] { 0, 1, 2, 3, 4, 5 };
            ViewBag.Gender = new SelectList(db.Customers.Select(x =>
x.Gender).Distinct());
            ViewBag.Marital_Status = new SelectList(db.Customers.Select(x =>
x.Marital_Status).Distinct());
            ViewBag.Education = new SelectList(db.Customers.Select(x =>
x.Education).Distinct());
            ViewBag.Occupation = new SelectList(db.Customers.Select(x =>
x.Occupation).Distinct());
            ViewBag.Home_Owner = new SelectList(db.Customers.Select(x =>
x.Home_Owner).Distinct());
            ViewBag.Cars = new SelectList(db.Customers.Select(x => x.Cars).Distinct());
            ViewBag.Commute_Distance = new SelectList(db.Customers.Select(x =>
x.Commute_Distance).Distinct());
            ViewBag.Region = new SelectList(db.Customers.Select(x =>
x.Region).Distinct());
            ViewBag.Children = new SelectList(Children);
            //requesting whatever user inserted
            string Commute_Distance = Request["Commute_Distance"].ToString();
            string Education = Request.Form["Education"].ToString();
            string Gender = Request.Form["Gender"].ToString();
            string Home_Owner = Request.Form["Home_Owner"].ToString();
            string Marital_Status = Request.Form["Marital_Status"].ToString();
            string Occupation = Request.Form["Occupation"].ToString();
            string Region = Request.Form["Region"].ToString();
```

```
            int getage = Convert.ToInt32(Request.Form["Age"]);
            int getcar = Convert.ToInt32(Request.Form["Cars"]);
            int getchild = Convert.ToInt32(Request.Form["Children"]);
            int getincome = Convert.ToInt32(Request.Form["Income"]);
            //starting the connection
            AdomdConnection con = new AdomdConnection("Data Source=WS\\SQLEXPRESS;
Catalog=ADMDataBase");
            con.Open();
            AdomdCommand cmd = new AdomdCommand();
            cmd.Connection = con;
            //quering the data base
            string s = @"Select flattened Category from BIKECOCCMM natural prediction
join
                                        (select '" + getage + @"' as [Age],
                                         '" + getcar + @"' as [Cars],
                                         '" + getchild + @" ' as [Children],
                                         '" + Commute_Distance + @"' as [Commute
Distance],
                                         '" + Education + @"' as [Education],
                                         '" + Gender + @"' as [Gender],
                                         '" + Home_Owner + @"' as [Home Owner],
                                         '" + getincome + @"' as [Income],
                                         '" + Marital_Status + @"' as [Marital
Status],
                                         '" + Occupation + @"' as [Occupation],
                                         '" + Region + @"' as [Region]) as t";
            cmd.CommandText = s;
            AdomdDataReader dr = cmd.ExecuteReader();
            string final_result = "";
            //returen the value from db, stor it and send it to view
            if (dr.Read())
            {
                if (dr[0] != null)
                {
                    final_result += dr[0].ToString();
                    ViewBag.final_result = dr[0].ToString();
                }
                dr.Close();
                con.Close();
            }
            return View();
        }
```

3- Get personalized recommendations

```
//on page-load
        public ActionResult Get_Recommendation()
        {
            //setting variables and send it to view
            int[] Children = new int[] { 0, 1, 2, 3, 4, 5 };
            ViewBag.Gender = new SelectList(db.Customers.Select(x =>
x.Gender).Distinct());
            ViewBag.Marital_Status = new SelectList(db.Customers.Select(x =>
x.Marital_Status).Distinct());
```

```
            ViewBag.Education = new SelectList(db.Customers.Select(x =>
x.Education).Distinct());
            ViewBag.Occupation = new SelectList(db.Customers.Select(x =>
x.Occupation).Distinct());
            ViewBag.Home_Owner = new SelectList(db.Customers.Select(x =>
x.Home_Owner).Distinct());
            ViewBag.Cars = new SelectList(db.Customers.Select(x => x.Cars).Distinct());
            ViewBag.Commute_Distance = new SelectList(db.Customers.Select(x =>
x.Commute_Distance).Distinct());
            ViewBag.Region = new SelectList(db.Customers.Select(x =>
x.Region).Distinct());
            ViewBag.Children = new SelectList(Children);


            return View();
        }
        //on post, or click
        [HttpPost, ActionName("Get_Recommendation")]
        public ActionResult Get_RecommendationPost()
        {

            int[] Children = new int[] { 0, 1, 2, 3, 4, 5 };
            ViewBag.Gender = new SelectList(db.Customers.Select(x =>
x.Gender).Distinct());
            ViewBag.Marital_Status = new SelectList(db.Customers.Select(x =>
x.Marital_Status).Distinct());
            ViewBag.Education = new SelectList(db.Customers.Select(x =>
x.Education).Distinct());
            ViewBag.Occupation = new SelectList(db.Customers.Select(x =>
x.Occupation).Distinct());
            ViewBag.Home_Owner = new SelectList(db.Customers.Select(x =>
x.Home_Owner).Distinct());
            ViewBag.Cars = new SelectList(db.Customers.Select(x => x.Cars).Distinct());
            ViewBag.Commute_Distance = new SelectList(db.Customers.Select(x =>
x.Commute_Distance).Distinct());
            ViewBag.Region = new SelectList(db.Customers.Select(x =>
x.Region).Distinct());
            ViewBag.Children = new SelectList(Children);
            //get the results from the user
            string Commute_Distance = Request["Commute_Distance"].ToString();
            string Education = Request.Form["Education"].ToString();
            string Gender = Request.Form["Gender"].ToString();
            string Home_Owner = Request.Form["Home_Owner"].ToString();
            string Marital_Status = Request.Form["Marital_Status"].ToString();
            string Occupation = Request.Form["Occupation"].ToString();
            string Region = Request.Form["Region"].ToString();
            int getage = Convert.ToInt32(Request.Form["Age"]);
            int getcar = Convert.ToInt32(Request.Form["Cars"]);
            int getchild = Convert.ToInt32(Request.Form["Children"]);
            int getincome = Convert.ToInt32(Request.Form["Income"]);
            //start connection
            AdomdConnection con = new AdomdConnection("Data Source=WS\\SQLEXPRESS;
Catalog=ADMDataBase");
            con.Open();
            AdomdCommand cmd = new AdomdCommand();
            cmd.Connection = con;
            //start query the DB
```

```
            string s = @"Select predict ([Product]) from BIKECOGRMM natural prediction
join
                                        (select '" + getage + @"' as [Age],
                                         '" + getcar + @"' as [Cars],
                                         '" + getchild + @" ' as [Children],
                                         '" + Commute_Distance + @"' as [Commute
Distance],
                                         '" + Education + @"' as [Education],
                                         '" + Gender + @"' as [Gender],
                                         '" + Home_Owner + @"' as [Home Owner],
                                         '" + getincome + @"' as [Income],
                                         '" + Marital_Status + @"' as [Marital
Status],
                                         '" + Occupation + @"' as [Occupation],
                                         '" + Region + @"' as [Region]) as t";
            cmd.CommandText = s;
            AdomdDataReader dr = cmd.ExecuteReader();
            string final_result = "";
            //returen the result
            if (dr.Read())
            {
                if (dr[0] != null)
                {
                    final_result += dr[0].ToString();
                    ViewBag.final_result = dr[0].ToString();
                }
                dr.Close();
                con.Close();
            }
            return View();
        }
```

ALL of these 3 parts was in customer's controller, the next 2 parts are in orders details controller.

4- Bought together

```
//on page-load
        public ActionResult Bought_Together()
        {
            //decalre a product list to save the chossen product inside it
            productlistitem listedproduct = new productlistitem();

            List<SelectListItem> listproducts = new List<SelectListItem>();
            //choose the products from DB
            foreach (Product choosen in db.Products.Distinct())
            {
                SelectListItem s = new SelectListItem()
                {
                    Text = choosen.Product1,
                    Value = choosen.Product1.ToString(),
                    Selected = false
                };
                listproducts.Add(s);
            }
```

```
            //inserting chossen products into the list
            listedproduct.chossproducts = listproducts;

            return View(listedproduct);
        }
        //on post use the list of products that user define
        [HttpPost, ActionName("Bought_Together")]
        public ActionResult Bought_Together(IEnumerable<string> chossenproducts)
        {
            //reseting output on each clcik
            TempData["output"] = "";
            string res = "";
            string[] splited = { "" };
            //sending the selected items to view to show it to user
            if (chossenproducts == null)
            {
                TempData["selected"] = "You Don't choose any item!";
            }
            else
            {
                res = string.Join(",", chossenproducts);
                TempData["selected"] = res;
            }
            splited = res.Split(',');
            //dealing with split list "can't use the form way, its MVC"
            for (int i = 0; i <= splited.Length-1; i++)
            {
                TempData["splited"] += splited[i];
            }
            //declaring input and out lists to be send to another function to returen
needed items
            List<string> input = new List<string>();
            List<string> output = new List<string>();
            //we need only 3 items to be shown
            int count = 3;
            //adding the selected item to input list
            foreach (string i in splited)
            {
                input.Add(i);
            }
            //sending the values to another function, and waiting for its output
            predict(input, output, count);
            //reciving the output and send it to view
            if (output!=null)
            {
                res = string.Join(",", output);

                TempData["output"] += res;
            }
            return RedirectToAction("Bought_Together");
        }
        //function to handel the query
        private void predict(List<string> input, List<string> output, int count)
        {
            //starting the connection
            AdomdConnection CON = new AdomdConnection("Data Source=WS\\SQLEXPRESS;
Catalog=ADMDataBase");
            CON.Open();
```

```
            AdomdCommand COM = CON.CreateCommand();
            //quering the DB
           string s = @"SELECT Flattened  PREDICT([Orders Details]," + count + @")
           FROM [BIKECOAITEMSMM]
            NATURAL PREDICTION JOIN
            (SELECT ( ";
            //count the items to produce valid query
            foreach (string x in input)
            {
                if (input.IndexOf(x) > 0)
                    s += " Union ";
                //skiping special carecters (only seen in "Women's"!!!!!!short
product..... comeon!)
                s += "Select '" + x.Replace("'", "''") + "' as [product]";
            }
            s += " )  AS [Orders Details]) As T;";
            COM.CommandText = s;

            AdomdDataReader DR = COM.ExecuteReader();
            //send the returend value to output
            while (DR.Read())
            {

                if (DR[0] != null)
                    output.Add(DR[0].ToString());
            }
            DR.Close();
            CON.Close();
        }
```

5- Will be

```
//time series on page load
        public ActionResult Will_Be()
        {
            //declaring calculations from 1-90 periods only
            int[] days = Enumerable.Range(1,90).ToArray();
            ViewBag.days = new SelectList(days);
            return View();
        }
        //on post
        [HttpPost, ActionName("Will_Be")]
        public ActionResult Will_Be(IEnumerable<string> chossenproducts)
        {
            int[] days = Enumerable.Range(1, 90).ToArray();
            ViewBag.days = new SelectList(days);
            int getdays = Convert.ToInt32(Request.Form["days"]);

            //reset the old results
            TempData["output"] = "";
            string res = "";
            //declaring list of products that returend from prediction function
            List<string> output = new List<string>();
            int count = getdays;
            //send the count of perdios and waiting for the output
```

```
            will_predict(output, count);
            //send the output to the view
            if (output != null)
            {
                res = string.Join(",", output);

                TempData["output"] += res;
            }
            return RedirectToAction("Will_Be");
        }
        //predicti function
        private void will_predict(List<string> output, int count)
        {
            //start connection
            AdomdConnection CON = new AdomdConnection("Data Source=WS\\SQLEXPRESS;
Catalog=ADMDataBase");
            CON.Open();

            AdomdCommand COM = CON.CreateCommand();
            //quering the database
            string s = @"SELECT Flattened  PredictTimeSeries([Prise Of Order]," + count +
@")
            as Forecast from BIKECOTSMM;";
            COM.CommandText = s;

            AdomdDataReader DR = COM.ExecuteReader();
            //setting the format of retuernd amount
            double shape = 0;
            string format = "";
            //send the output formated clearly to the main function to be views
            while (DR.Read())
            {
                if (DR[0] != null)
                shape = Convert.ToDouble(DR[1]);
                format = shape.ToString("F2");
                output.Add(format);
            }
            DR.Close();
            CON.Close();
        }
```

Done.