MySQL Assignment

Name: Akash Choudhuri
Roll: 2019D014.

Showing databases/Schema present in MySQL workbench:

```
show databases;
```

Creating a database named as wella:

```
create database wella;
use wella;
```

Create Table:

```
CREATE TABLE `wella_brand` (

  `brand_id` varchar(64) NOT NULL,

  `brand_name` varchar(255) NOT NULL,

  `brand_image` varchar(255) DEFAULT NULL,

  `created_at` timestamp NOT NULL,

  `updated_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`brand_id`)

);
```

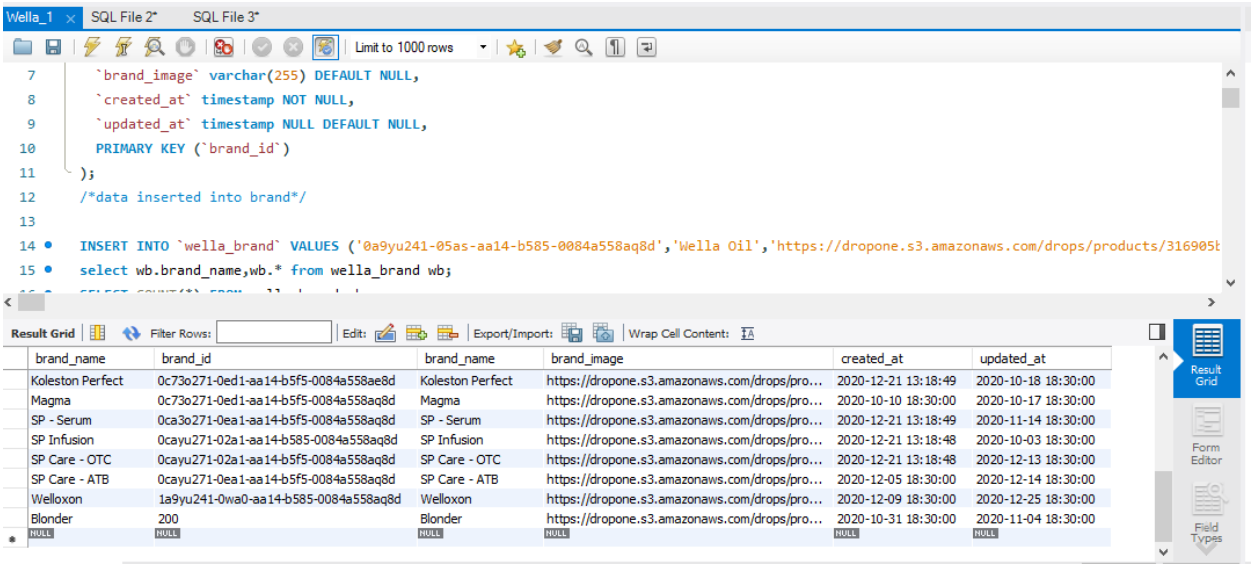
Data inserted into brand:

```
INSERT INTO `wella_brand` VALUES ('0a9yu241-05as-aa14-b585-0084a558aq8d','Wella
Oil','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-ae59-
4ccac73ca34d/mylogo.jpg','2020-10-31 18:30:00','2020-11-04 18:30:00'),('0a9yu241-0wa0-aa14-b585-
0084a558aq8d','Wellaplex','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-
ae59-4ccac73ca34d/mylogo.jpg','2020-11-09 18:30:00','2020-11-14 18:30:00'),('0a9yu241-0was-aa14-
b585-0084a558aq8d','Wella Style','https://dropone.s3.amazonaws.com/drops/products/316905b2-
79a7-4478-ae59-4ccac73ca34d/mylogo.jpg','2020-12-21 11:20:06','2020-11-11 18:30:00'),('0a9yu271-
02a0-aa14-b585-0084a558aq8d','Wella Care -
OTC','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-ae59-
4ccac73ca34d/mylogo.jpg','2020-12-21 11:20:06','2020-11-09 18:30:00'),('0a9yu271-05a0-aa14-b585-
0084a558aq8d','Wella Care - Serum','https://dropone.s3.amazonaws.com/drops/products/316905b2-
79a7-4478-ae59-4ccac73ca34d/mylogo.jpg','2020-12-21 11:20:06','2020-11-10 18:30:00'),('0a9yu271-
```

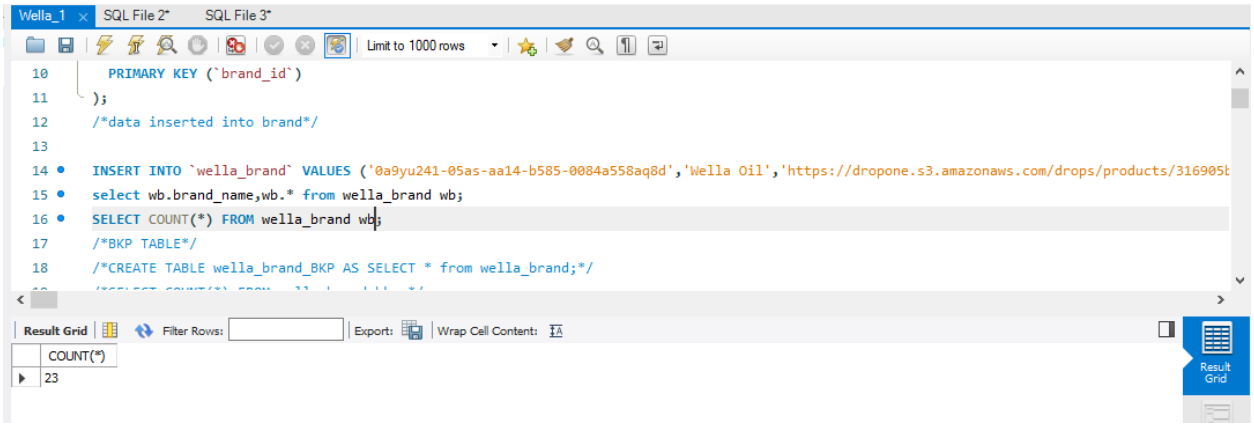
05as-aa14-b585-0084a558aq8d','Wella
Fusion','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-ae59-
4ccac73ca34d/mylogo.jpg','2020-12-21 11:20:06','2020-12-10 18:30:00'),('0a9yu2e1-05as-aa14-b585-
0084a558aq8d','Wella Helios','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-
4478-ae59-4ccac73ca34d/mylogo.jpg','2020-12-21 11:20:06','2020-12-04 18:30:00'),('0aayu271-02a0-
aa14-b585-0084a558aq8d','Wella Care -
ATB','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-ae59-
4ccac73ca34d/mylogo.jpg','2020-10-19 18:30:00','2020-10-28 18:30:00'),('0aayu271-02a1-aa14-b585-
0084a558aq8d','SP Luxe','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-
ae59-4ccac73ca34d/mylogo.jpg','2020-09-19 18:30:00','2020-10-04 18:30:00'),('0c737271-0ed1-4c14-
b5f5-9b84a558ae8c','Colortouch','https://dropone.s3.amazonaws.com/drops/products/316905b2-
79a7-4478-ae59-4ccac73ca34d/mylogo.jpg','2020-12-21 13:18:48','2020-11-04 18:30:00'),('0c737271-
0ed1-4c14-b5f5-
9b84a558ae8d','Illumina','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-
ae59-4ccac73ca34d/mylogo.jpg','2020-10-10 18:30:00','2020-11-27 18:30:00'),('0c737271-0ed1-4c14-
b5f5-9b84f558ae8c','Blondor','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-
4478-ae59-4ccac73ca34d/mylogo.jpg','2020-11-14 18:30:00','2020-11-15 18:30:00'),('0c737271-0ed1-
aa14-b5f5-0084a558ae8d','Invigo-ATB','https://dropone.s3.amazonaws.com/drops/products/316905b2-
79a7-4478-ae59-4ccac73ca34d/mylogo.jpg','2020-11-13 18:30:00','2020-11-18 18:30:00'),('0c737271-
0ed1-aa14-b5f5-
9b84a558ae8d','Invigo','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-
ae59-4ccac73ca34d/mylogo.jpg','2020-11-10 18:30:00','2020-11-27 18:30:00'),('0c73o271-0ea1-aa14-
b5f5-0084a558aq8d','Seal','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-
ae59-4ccac73ca34d/mylogo.jpg','2020-12-21 13:18:49','2020-10-14 18:30:00'),('0c73o271-0ed1-aa14-
b5f5-0084a558ae8d','Koleston
Perfect','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-ae59-
4ccac73ca34d/mylogo.jpg','2020-12-21 13:18:49','2020-10-18 18:30:00'),('0c73o271-0ed1-aa14-b5f5-
0084a558aq8d','Magma','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-
ae59-4ccac73ca34d/mylogo.jpg','2020-10-10 18:30:00','2020-10-17 18:30:00'),('0ca3o271-0ea1-aa14-
b5f5-0084a558aq8d','SP - Serum','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-
4478-ae59-4ccac73ca34d/mylogo.jpg','2020-12-21 13:18:49','2020-11-14 18:30:00'),('0cayu271-02a1-
aa14-b585-0084a558aq8d','SP
Infusion','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-ae59-
4ccac73ca34d/mylogo.jpg','2020-12-21 13:18:48','2020-10-03 18:30:00'),('0cayu271-02a1-aa14-b5f5-
0084a558aq8d','SP Care - OTC','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-
4478-ae59-4ccac73ca34d/mylogo.jpg','2020-12-21 13:18:48','2020-12-13 18:30:00'),('0cayu271-0ea1-
aa14-b5f5-0084a558aq8d','SP Care -
ATB','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-ae59-
4ccac73ca34d/mylogo.jpg','2020-12-05 18:30:00','2020-12-14 18:30:00'),('1a9yu241-0wa0-aa14-b585-
0084a558aq8d','Welloxon','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-
ae59-4ccac73ca34d/mylogo.jpg','2020-12-09 18:30:00','2020-12-25

18:30:00'),('200','Blonder','https://dropone.s3.amazonaws.com/drops/products/316905b2-79a7-4478-ae59-4ccac73ca34d','2020-10-31 18:30:00','2020-11-04 18:30:00');

select wb.brand_name,wb.* from wella_brand wb;



SELECT COUNT(*) FROM wella_brand wb;



Backup Table:

/*CREATE TABLE wella_brand_BKP AS SELECT * from wella_brand;*/

/*SELECT COUNT(*) FROM wella_brand_bkp;*/

/*wella_categbrand_nameories*/

drop table wella_brand_bkp;

```
SELECT COUNT(*) FROM wella_brand_bkp;
```

Create Another Table:

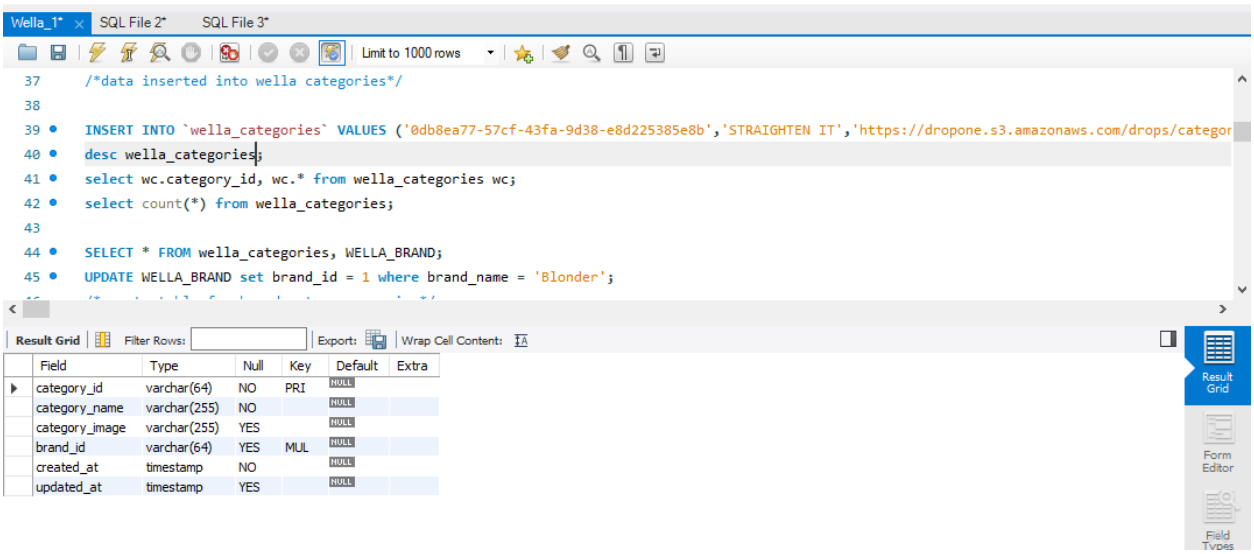
```
CREATE TABLE `wella_categories` (

  `category_id` varchar(64) NOT NULL,

  `category_name` varchar(255) NOT NULL,

  `category_image` varchar(255) DEFAULT NULL,

  `brand_id` varchar(64) DEFAULT NULL,

  `created_at` timestamp NOT NULL,

  `updated_at` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`category_id`),

  KEY `brand_id` (`brand_id`),

  CONSTRAINT `wella_categories_ibfk_1` FOREIGN KEY (`brand_id`) REFERENCES `wella_brand`
(`brand_id`)

);
```

Data insertion into wella_categories:

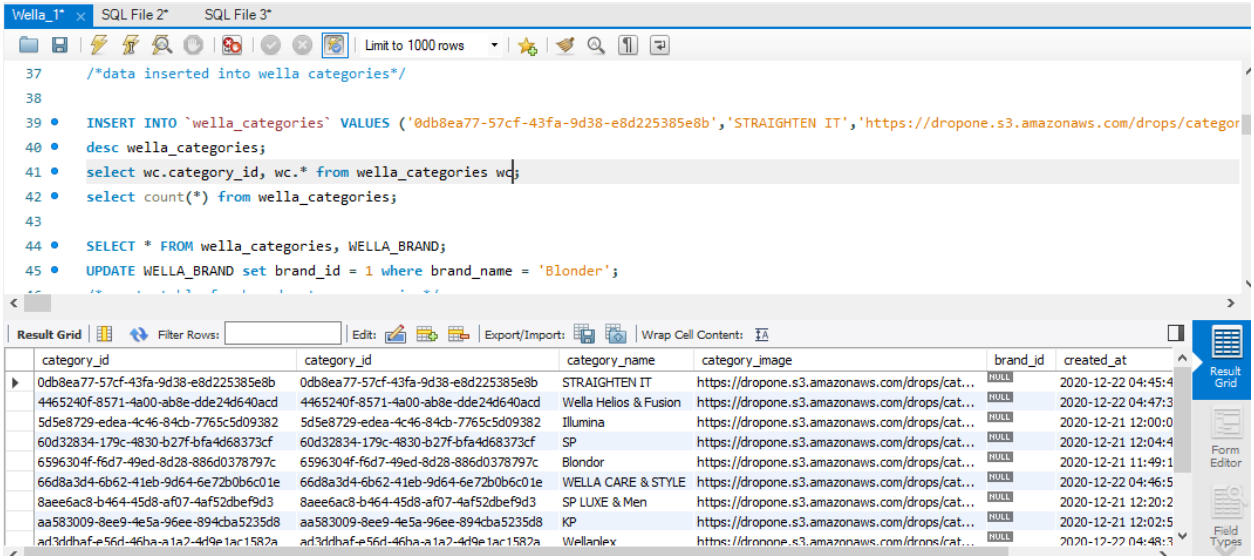
```
INSERT INTO `wella_categories` VALUES ('0db8ea77-57cf-43fa-9d38-e8d225385e8b','STRAIGHTEN
IT','https://dropone.s3.amazonaws.com/drops/categories/0db8ea77-57cf-43fa-9d38-
e8d225385e8b/5.jpg',NULL,'2020-12-22 04:45:45','2020-12-22 04:45:45'),('4465240f-8571-4a00-ab8e-
dde24d640acd','Wella Helios &
Fusion','https://dropone.s3.amazonaws.com/drops/categories/4465240f-8571-4a00-ab8e-
dde24d640acd/10.jpg',NULL,'2020-12-22 04:47:32','2020-12-22 04:47:32'),('5d5e8729-edea-4c46-84cb-
7765c5d09382','Illumina','https://dropone.s3.amazonaws.com/drops/categories/5d5e8729-edea-4c46-
84cb-7765c5d09382/3.jpg',NULL,'2020-12-21 12:00:08','2020-12-21 12:00:08'),('60d32834-179c-4830-
b27f-bfa4d68373cf','SP','https://dropone.s3.amazonaws.com/drops/categories/60d32834-179c-4830-
b27f-bfa4d68373cf/1.jpg',NULL,'2020-12-21 12:04:48','2020-12-21 12:04:48'),('6596304f-f6d7-49ed-
8d28-886d0378797c','Blondor','https://dropone.s3.amazonaws.com/drops/categories/6596304f-f6d7-
49ed-8d28-886d0378797c/1.jpg',NULL,'2020-12-21 11:49:13','2020-12-21 11:49:14'),('66d8a3d4-6b62-
41eb-9d64-6e72b0b6c01e','WELLA CARE &
STYLE','https://dropone.s3.amazonaws.com/drops/categories/66d8a3d4-6b62-41eb-9d64-
```

6e72b0b6c01e/12.jpg',NULL,'2020-12-22 04:46:52','2020-12-22 04:46:52'),('8aee6ac8-b464-45d8-af07-4af52dbef9d3','SP LUXE & Men','https://dropone.s3.amazonaws.com/drops/categories/8aee6ac8-b464-45d8-af07-4af52dbef9d3/6.jpg',NULL,'2020-12-21 12:20:26','2020-12-21 12:20:26'),('aa583009-8ee9-4e5a-96ee-894cba5235d8','KP','https://dropone.s3.amazonaws.com/drops/categories/aa583009-8ee9-4e5a-96ee-894cba5235d8/4.jpg',NULL,'2020-12-21 12:02:55','2020-12-21 12:02:55'),('ad3ddbaf-e56d-46ba-a1a2-
4d9e1ac1582a','Wellaplex','https://dropone.s3.amazonaws.com/drops/categories/ad3ddbaf-e56d-46ba-a1a2-4d9e1ac1582a/7.jpg',NULL,'2020-12-22 04:48:34','2020-12-22 04:48:34'),('ba9d5930-d1ba-499c-89cf-
a56d7da39fd2','WELLOXON','https://dropone.s3.amazonaws.com/drops/categories/ba9d5930-d1ba-499c-89cf-a56d7da39fd2/12.jpg',NULL,'2020-12-22 04:49:00','2020-12-22 04:49:00'),('cdcd5926-c830-4479-9736-21997e1f4ba1','WELLA
OIL','https://dropone.s3.amazonaws.com/drops/categories/cdcd5926-c830-4479-9736-21997e1f4ba1/11.jpg',NULL,'2020-12-22 04:48:03','2020-12-22 04:48:03'),('f54c2e1d-9f6c-4f49-87d8-2d4843d5c575','CT & Magma','https://dropone.s3.amazonaws.com/drops/categories/f54c2e1d-9f6c-4f49-87d8-2d4843d5c575/2.jpg',NULL,'2020-12-21 11:54:15','2020-12-21 11:54:15');

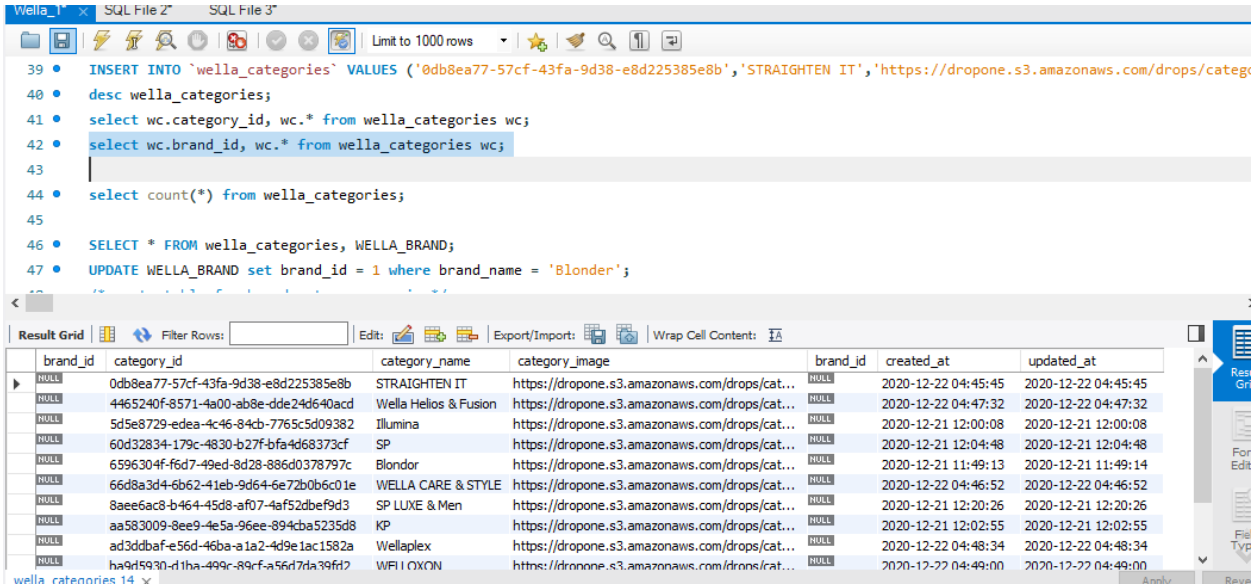
desc wella_categories;



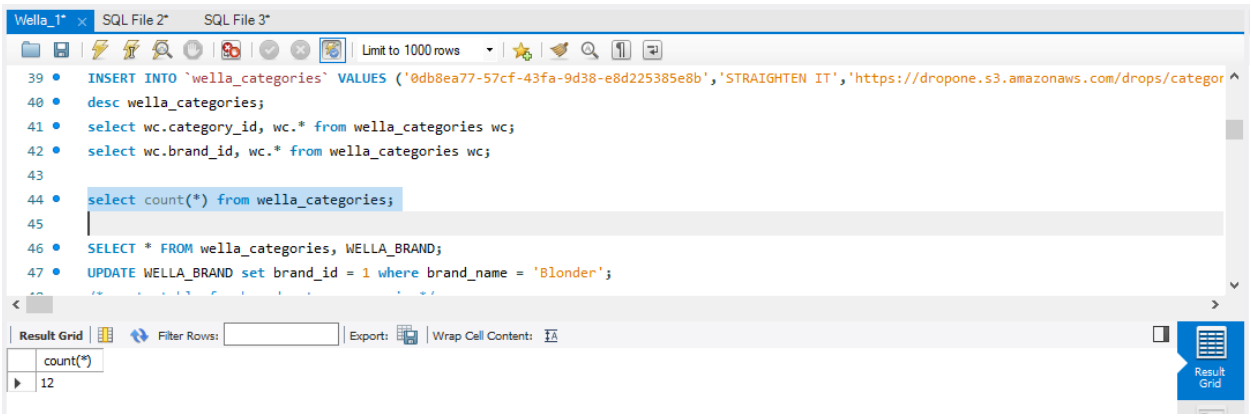
select wc.category_id, wc.* from wella_categories wc;



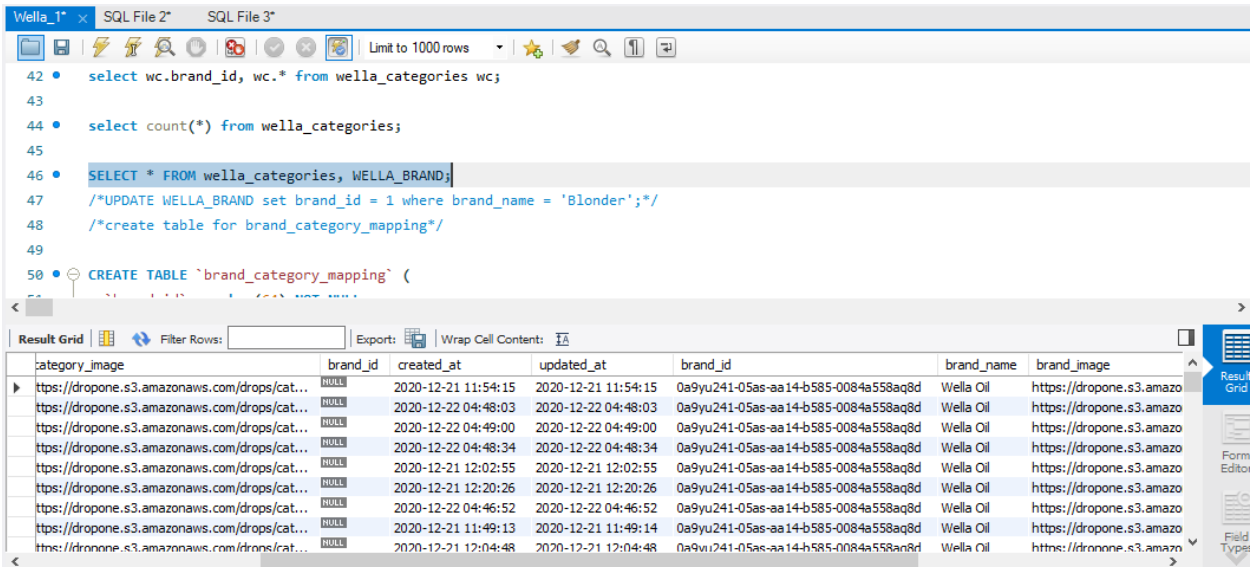
Select wc.brand_id, wc.* from wella_categories wc;



select count(*) from wella_categories;



SELECT * FROM wella_categories, WELLA_BRAND;



Create table for brand_category_mapping:

CREATE TABLE `brand_category_mapping` (

`brand_id` varchar(64) NOT NULL,

`category_id` varchar(64) NOT NULL,

KEY `brand_id` (`brand_id`),

KEY `category_id` (`category_id`),

CONSTRAINT `brand_category_mapping_ibfk_1` FOREIGN KEY (`brand_id`) REFERENCES `wella_brand` (`brand_id`),

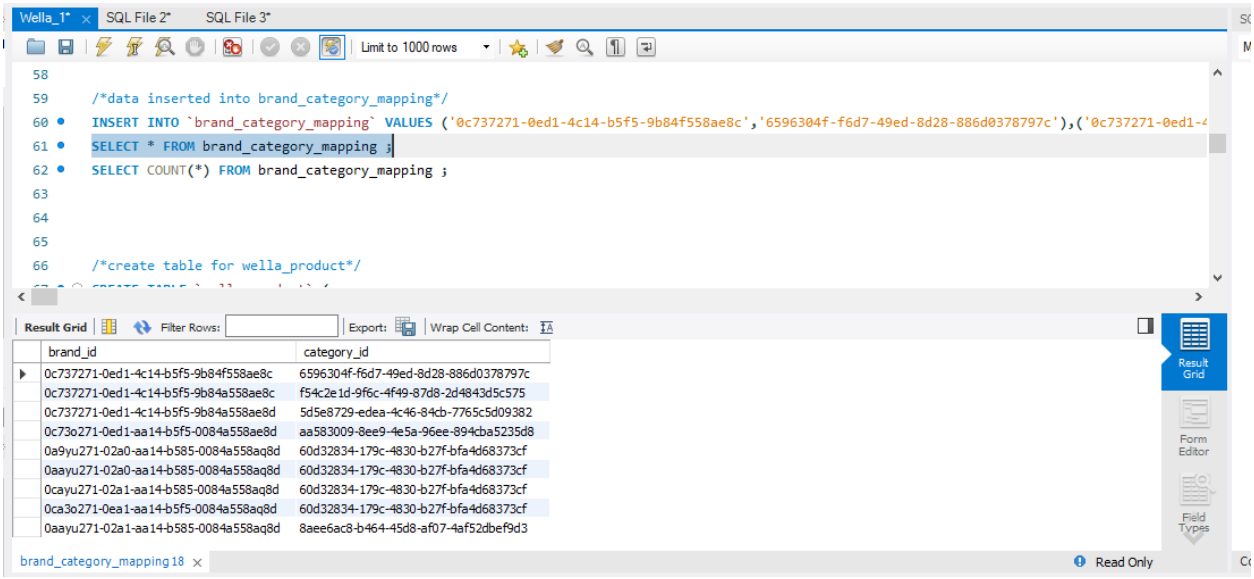
CONSTRAINT `brand_category_mapping_ibfk_2` FOREIGN KEY (`category_id`) REFERENCES `wella_categories` (`category_id`)

);

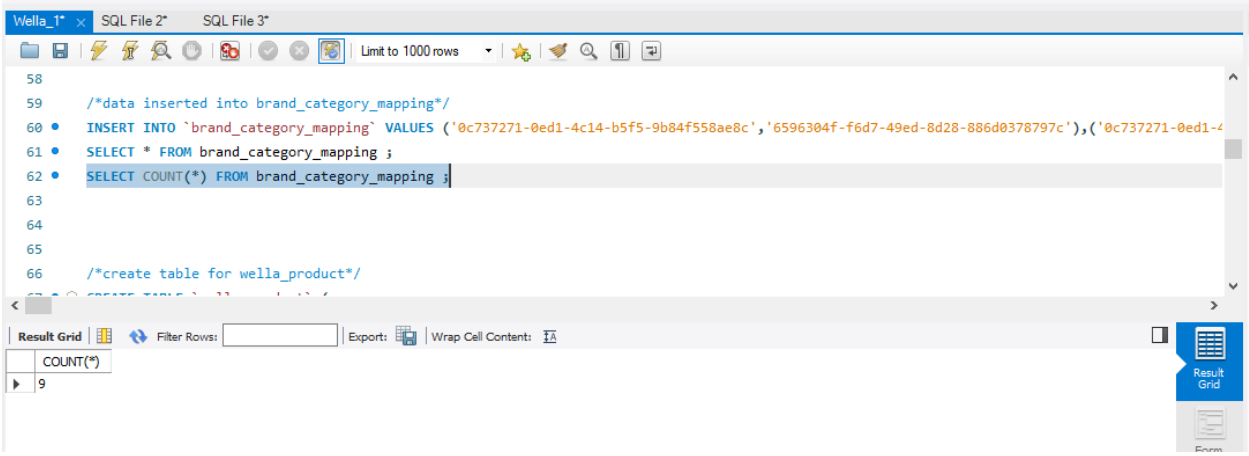
/*data inserted into brand_category_mapping*/

INSERT INTO `brand_category_mapping` VALUES ('0c737271-0ed1-4c14-b5f5-9b84f558ae8c','6596304f-f6d7-49ed-8d28-886d0378797c'),('0c737271-0ed1-4c14-b5f5-9b84a558ae8c','f54c2e1d-9f6c-4f49-87d8-2d4843d5c575'),('0c737271-0ed1-4c14-b5f5-9b84a558ae8d','5d5e8729-edea-4c46-84cb-7765c5d09382'),('0c73o271-0ed1-aa14-b5f5-0084a558ae8d','aa583009-8ee9-4e5a-96ee-894cba5235d8'),('0a9yu271-02a0-aa14-b585-0084a558aq8d','60d32834-179c-4830-b27f-bfa4d68373cf'),('0aayu271-02a0-aa14-b585-0084a558aq8d','60d32834-179c-4830-b27f-bfa4d68373cf'),('0cayu271-02a1-aa14-b585-0084a558aq8d','60d32834-179c-4830-b27f-bfa4d68373cf'),('0ca3o271-0ea1-aa14-b5f5-0084a558aq8d','60d32834-179c-4830-b27f-bfa4d68373cf'),('0aayu271-02a1-aa14-b585-0084a558aq8d','8aee6ac8-b464-45d8-af07-4af52dbef9d3');

SELECT * FROM brand_category_mapping ;



SELECT COUNT(*) FROM brand_category_mapping ;



Create table for wella_product:

```
CREATE TABLE `wella_product` (
  `product_id` varchar(64) NOT NULL,
  `product_code` bigint DEFAULT NULL,
  `product_name` varchar(255) DEFAULT NULL,
  `product_image` varchar(1000) DEFAULT NULL,
  `mrp` double DEFAULT NULL,
  `brand_id` varchar(64) DEFAULT NULL,
  `category_id` varchar(64) DEFAULT NULL,
  `created_at` timestamp NOT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`product_id`),
  KEY `brand_id` (`brand_id`),
  KEY `category_id` (`category_id`),
  CONSTRAINT `wella_product_ibfk_1` FOREIGN KEY (`brand_id`) REFERENCES `wella_brand`
(`brand_id`),
```

CONSTRAINT `wella_product_ibfk_2` FOREIGN KEY (`category_id`) REFERENCES `wella_categories` (`category_id`)

);

Data insertion into wella_product:

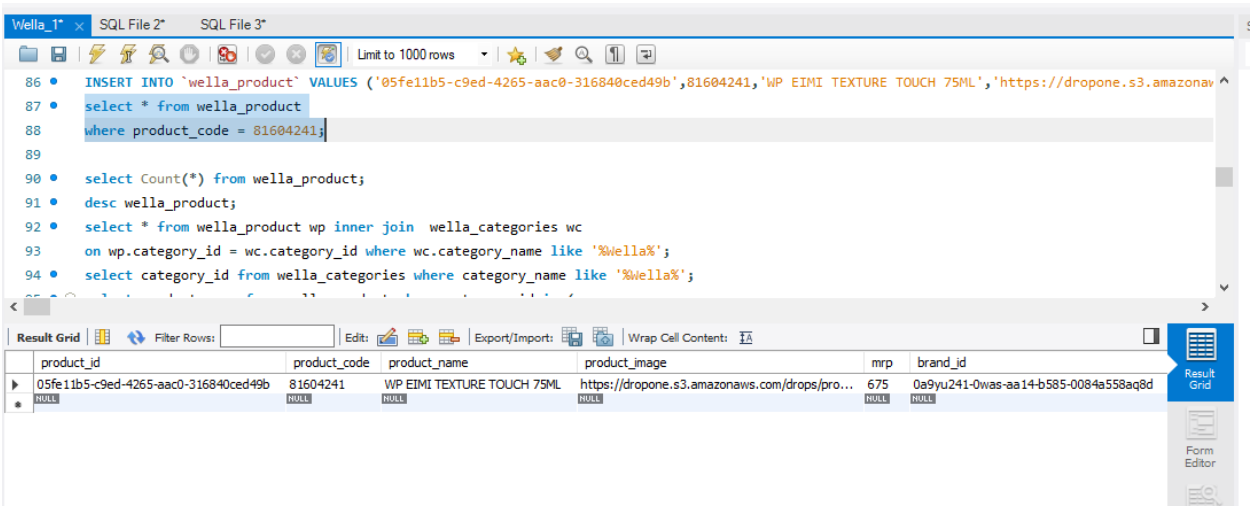
INSERT INTO `wella_product` VALUES ('05fe11b5-c9ed-4265-aac0-316840ced49b',81604241,'WP EIMI TEXTURE TOUCH 75ML','https://dropone.s3.amazonaws.com/drops/products/05fe11b5-c9ed-4265-aac0-316840ced49b/9.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:01:29','2020-12-22 05:01:29'),('077f837a-e8dc-4986-8fcb-be88c6850af1',81604246,'WP EIMI PEARL STYLER 150ML','https://dropone.s3.amazonaws.com/drops/products/077f837a-e8dc-4986-8fcb-be88c6850af1/5.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:07:18','2020-12-22 05:07:18'),('16d9f4dc-505e-444f-a8aa-a09fb0969f34',81604247,'WP EIMI RUGGED TEXTURE 75ML','https://dropone.s3.amazonaws.com/drops/products/16d9f4dc-505e-444f-a8aa-a09fb0969f34/3.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:08:09','2020-12-22 05:08:09'),('39ff1cf5-9505-4177-b900-7ace3939e98d',81572095,'BLOND Extra Cool 150G','https://dropone.s3.amazonaws.com/drops/products/39ff1cf5-9505-4177-b900-7ace3939e98d/8.jpg',825,'0c737271-0ed1-4c14-b5f5-9b84f558ae8c','6596304f-f6d7-49ed-8d28-886d0378797c','2020-12-22 05:16:55','2020-12-22 05:16:55'),('3e46693f-abb2-468e-834a-789029f2555f',81649974,'WP EIMI MISTIFY ME 300ML','https://dropone.s3.amazonaws.com/drops/products/3e46693f-abb2-468e-834a-789029f2555f/4.jpg',600,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 04:55:19','2020-12-22 04:55:19'),('4bd7dcc8-ecac-41b3-88e5-946fd0e4e4cc',81604233,'WP EIMI DRY ME 180ml','https://dropone.s3.amazonaws.com/drops/products/4bd7dcc8-ecac-41b3-88e5-946fd0e4e4cc/2.jpg',900,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 04:53:35','2020-12-22 04:53:35'),('58631f0e-de61-4637-80af-1624e93a41c0',81604243,'WP EIMI THERMAL IMAGE 150ML','https://dropone.s3.amazonaws.com/drops/products/58631f0e-de61-4637-80af-1624e93a41c0/11.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:04:18','2020-12-22 05:04:18'),('65434e19-8578-4629-a78b-ea71dc78414e',81604249,'WP EIMI SCULPT FORCE 125ML','https://dropone.s3.amazonaws.com/drops/products/65434e19-8578-4629-a78b-ea71dc78414e/6.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:09:33','2020-12-22 05:09:33'),('6563dba7-646a-452c-ac80-555712ec6ef8',81572097,'BLOND Soft BL Cream 200G','https://dropone.s3.amazonaws.com/drops/products/6563dba7-646a-452c-ac80-

555712ec6ef8/6.jpg',825,'0c737271-0ed1-4c14-b5f5-9b84f558ae8c','6596304f-f6d7-49ed-8d28-886d0378797c','2020-12-22 05:15:45','2020-12-22 05:15:45'),('6ad5390d-9206-42b1-a9f5-6e5e0118298a',81604231,'WP BOOST BOUNDS 300ML','https://dropone.s3.amazonaws.com/drops/products/6ad5390d-9206-42b1-a9f5-6e5e0118298a/10.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 04:52:46','2020-12-22 04:52:46'),('7104b69c-2d26-4221-9fe8-425aa13b8aa5',99350046940,'BLOND PLEX Powder 400G','https://dropone.s3.amazonaws.com/drops/products/7104b69c-2d26-4221-9fe8-425aa13b8aa5/6.jpg',1350,'0c737271-0ed1-4c14-b5f5-9b84f558ae8c','6596304f-f6d7-49ed-8d28-886d0378797c','2020-12-21 13:11:51','2020-12-21 13:11:52'),('a0bf3259-820a-4870-bbd6-3cad48dec836',81604251,'WP EIMI SUGAR LIFT 150ML','https://dropone.s3.amazonaws.com/drops/products/a0bf3259-820a-4870-bbd6-3cad48dec836/7.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:10:38','2020-12-22 05:10:38'),('b2497754-7d84-42b4-b645-706796c0c84e',81604238,'WP EIMI VELVET AMP 50ML','https://dropone.s3.amazonaws.com/drops/products/b2497754-7d84-42b4-b645-706796c0c84e/5.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 04:59:19','2020-12-22 04:59:19'),('b36d3779-9024-4912-ba03-2054cf4ee3ed',81607465,'WP EIMI JUST BRILLIANT 75ML','https://dropone.s3.amazonaws.com/drops/products/b36d3779-9024-4912-ba03-2054cf4ee3ed/9.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:12:00','2020-12-22 05:12:00'),('d80e6620-3160-42e5-a920-7cf952ea31f1',81604232,'WP EXTRA VOLUME 300Ml','https://dropone.s3.amazonaws.com/drops/products/d80e6620-3160-42e5-a920-7cf952ea31f1/9.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 04:52:02','2020-12-22 04:52:02'),('d8e705b1-dfdf-46be-8717-3d3e88f87111',99350052364,'BLOND Multi BL Pow 400G','https://dropone.s3.amazonaws.com/drops/products/d8e705b1-dfdf-46be-8717-3d3e88f87111/8.jpg',1250,'0c737271-0ed1-4c14-b5f5-9b84f558ae8c','6596304f-f6d7-49ed-8d28-886d0378797c','2020-12-22 05:14:59','2020-12-22 05:14:59'),('e11fcbdb-e269-476b-be8b-81e9dcdde34b',81604242,'WP EIMI OCEAN SPRITZ 150ML','https://dropone.s3.amazonaws.com/drops/products/e11fcbdb-e269-476b-be8b-81e9dcdde34b/10.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:02:25','2020-12-22 05:02:25'),('eef5ee17-2ad3-42b0-b546-2d8b696c01d2',81604240,'WP EIMI FLOWING FORM 100ML','https://dropone.s3.amazonaws.com/drops/products/eef5ee17-2ad3-42b0-b546-2d8b696c01d2/6.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:00:23','2020-12-22 05:00:23'),('ef331872-204d-49df-a82f-158093540e4f',81604244,'WP EIMI SHAPE SHIFT 150ML','https://dropone.s3.amazonaws.com/drops/products/ef331872-204d-49df-a82f-158093540e4f/12.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:05:12','2020-12-22 05:05:12'),('f5ea2b98-ae27-41df-bd46-

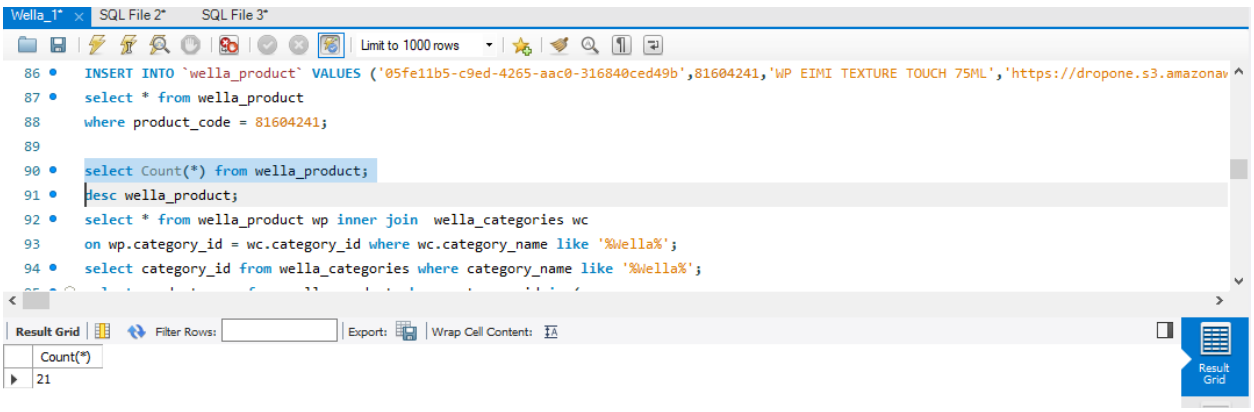
6dfe789039a4',81604245,'WP EIMI BOLD MOVE 150ML','https://dropone.s3.amazonaws.com/drops/products/f5ea2b98-ae27-41df-bd46-6dfe789039a4/13.jpg',675,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 05:06:18','2020-12-22 05:06:18'),('f7f94575-f00b-4a69-b713-cf62e24f44aa',81604235,'WP EIMI PERFECT ME 100ml','https://dropone.s3.amazonaws.com/drops/products/f7f94575-f00b-4a69-b713-cf62e24f44aa/3.jpg',900,'0a9yu241-0was-aa14-b585-0084a558aq8d','66d8a3d4-6b62-41eb-9d64-6e72b0b6c01e','2020-12-22 04:54:23','2020-12-22 04:54:23');

select * from wella_product

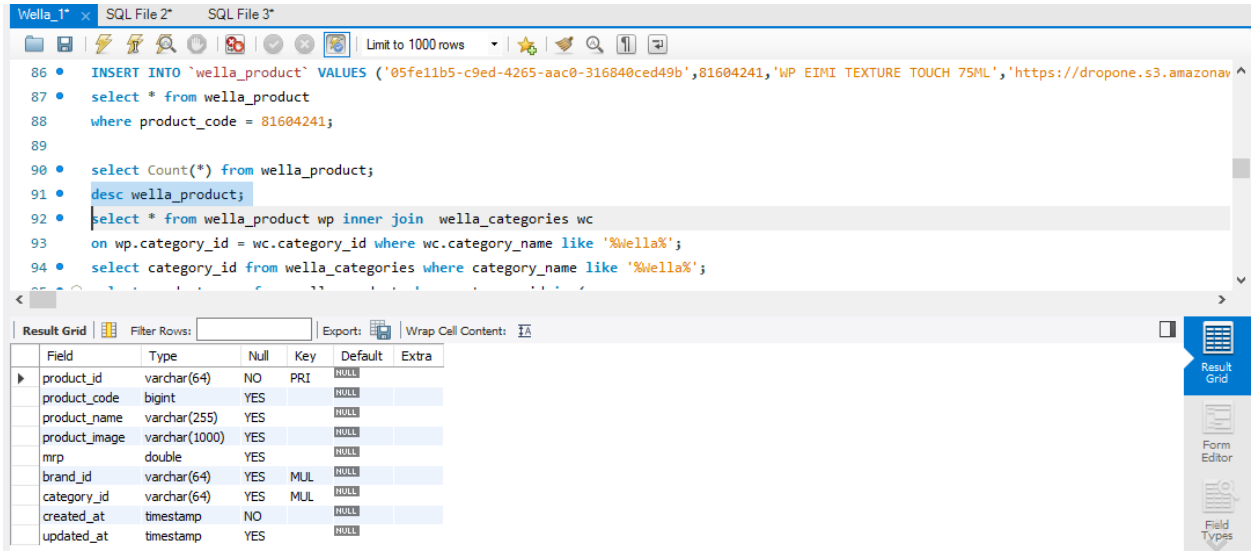
where product_code = 81604241;



select Count(*) from wella_product;

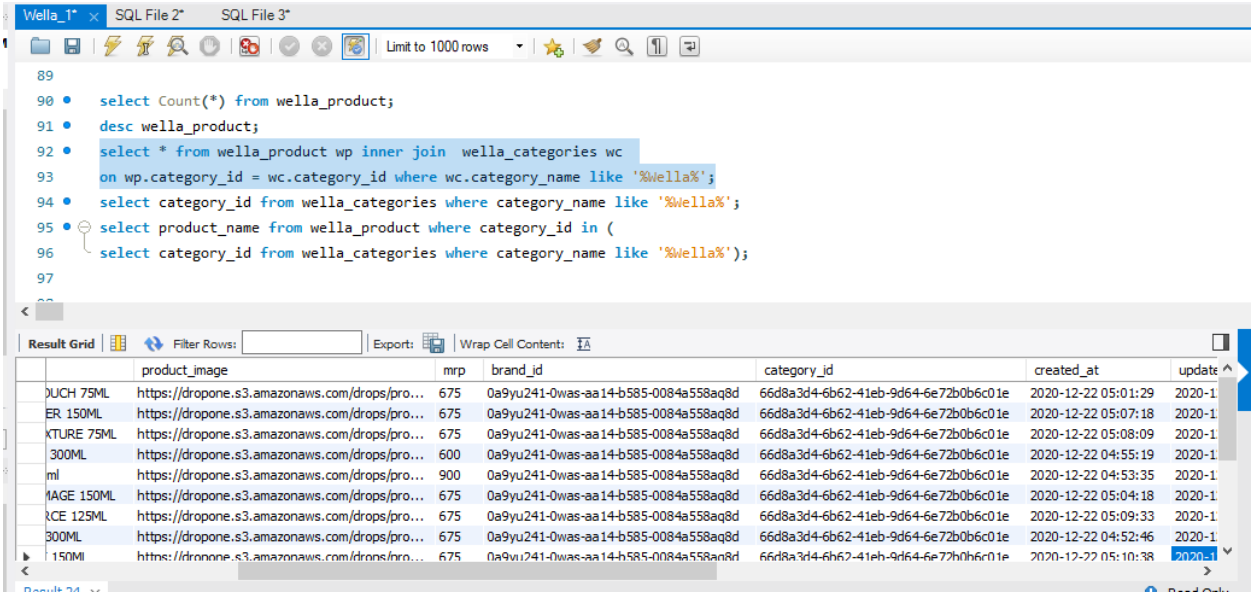


desc wella_product;

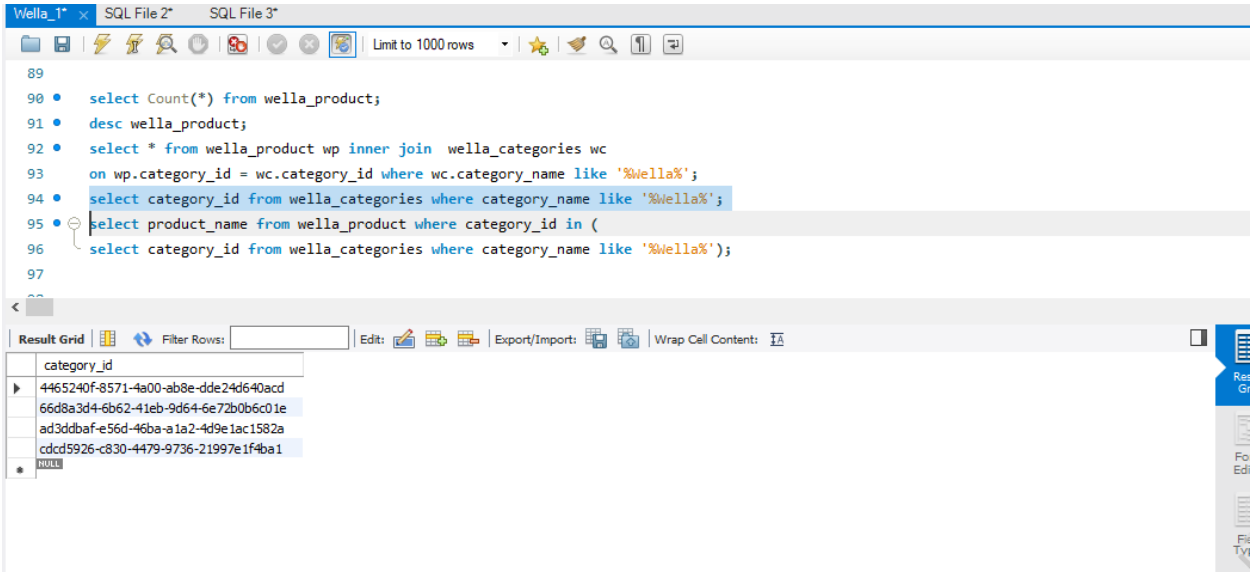


select * from wella_product wp inner join wella_categories wc

on wp.category_id = wc.category_id where wc.category_name like '%Wella%';

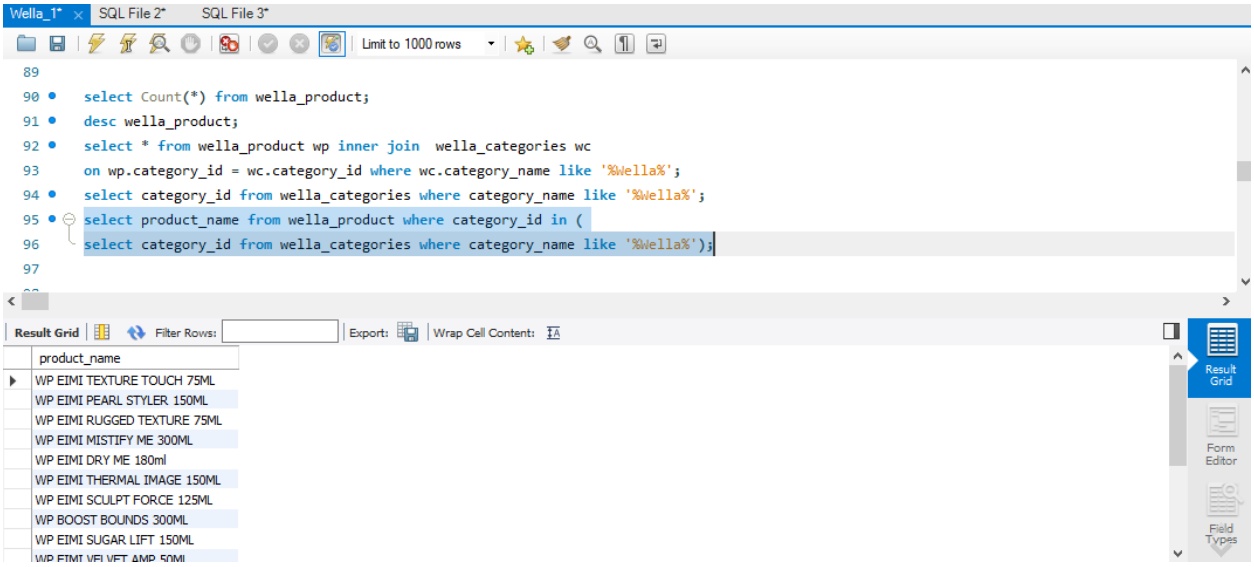


select category_id from wella_categories where category_name like '%Wella%';



select product_name from wella_product where category_id in (

select category_id from wella_categories where category_name like '%Wella%');



Create table for wella_userdetails:

CREATE TABLE `wella_userdetails` (

`user_id` varchar(64) NOT NULL,

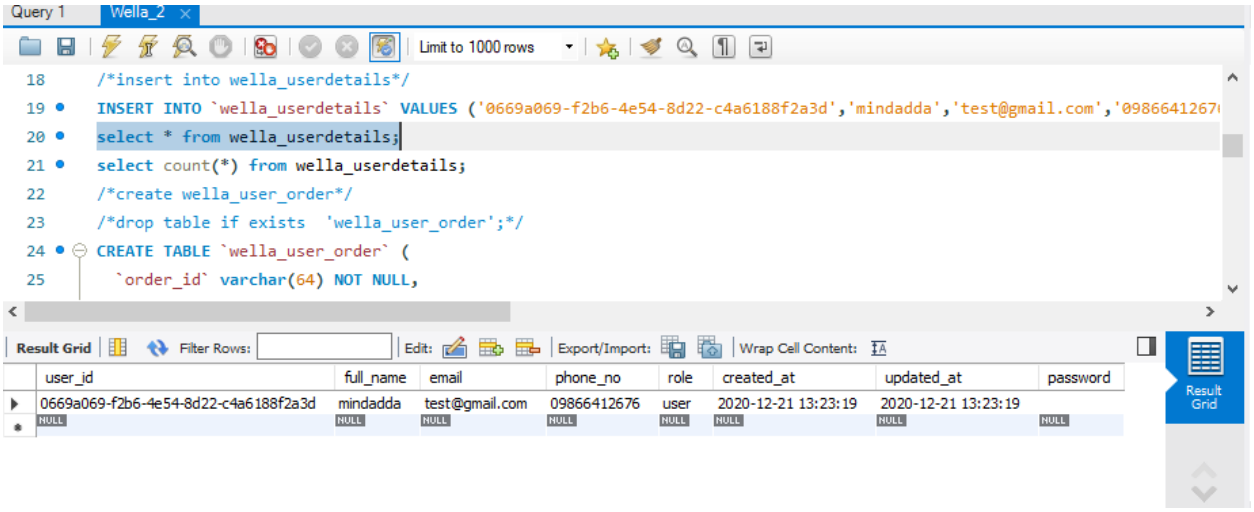
`full_name` varchar(255) NOT NULL,

```
 `email` varchar(255) NOT NULL,
 `phone_no` varchar(15) NOT NULL,
 `role` varchar(10) NOT NULL,
 `created_at` timestamp NOT NULL,
 `updated_at` timestamp NULL DEFAULT NULL,
 `password` varchar(255) NOT NULL,
 PRIMARY KEY (`user_id`),
 UNIQUE KEY `email` (`email`),
 UNIQUE KEY `phone_no` (`phone_no`)
);
```

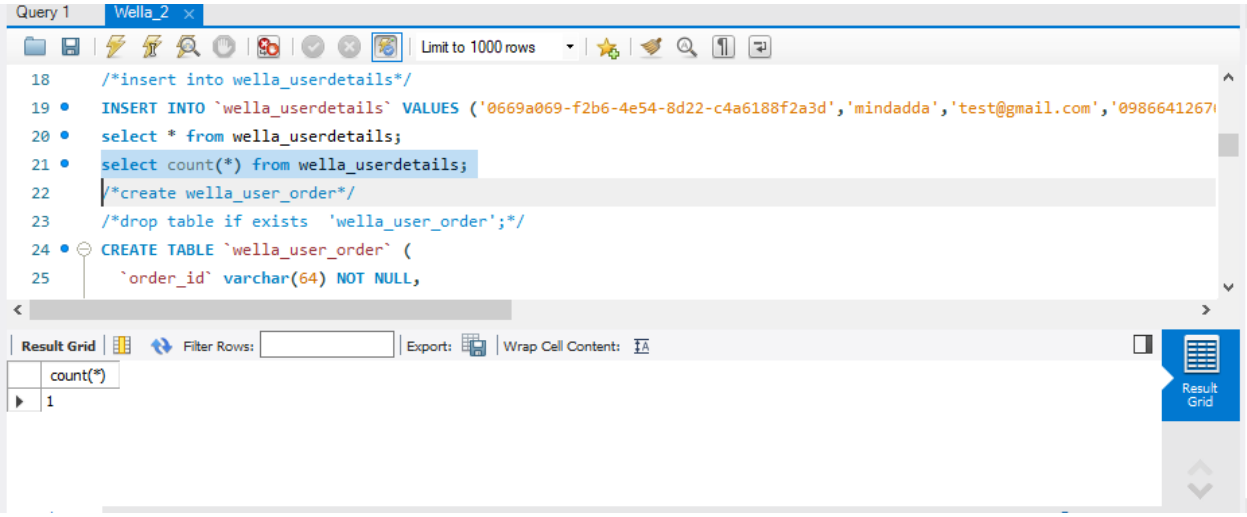
Insert into wella_userdetails:

```
INSERT INTO `wella_userdetails` VALUES ('0669a069-f2b6-4e54-8d22-c4a6188f2a3d','mindadda','test@gmail.com','09866412676','user','2020-12-21 13:23:19','2020-12-21 13:23:19','');
```

```
select * from wella_userdetails;
```




```
select count(*) from wella_userdetails;
```



Create wella_user_order:

```
CREATE TABLE `wella_user_order` (

 `order_id` varchar(64) NOT NULL,

 `total_price` double DEFAULT NULL,

 `bill_discount` double DEFAULT NULL,

 `final_bill_price` double DEFAULT NULL,

 `no_of_items` int DEFAULT NULL,

 `order_status` varchar(64) DEFAULT NULL,

 `user_id` varchar(64) DEFAULT NULL,



 `created_at` timestamp NOT NULL,

 `updated_at` timestamp NULL DEFAULT NULL,

 PRIMARY KEY (`order_id`),

 KEY `user_id` (`user_id`),

 CONSTRAINT `wella_user_order_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `wella_userdetails`
(`user_id`)

);
```

For adding extra coloumn into wella_user _order:

```
alter table wella_user_order add delivered_at timestamp;


/*insert into wella_user_order*/


/*insert into wella_user_order*/
```

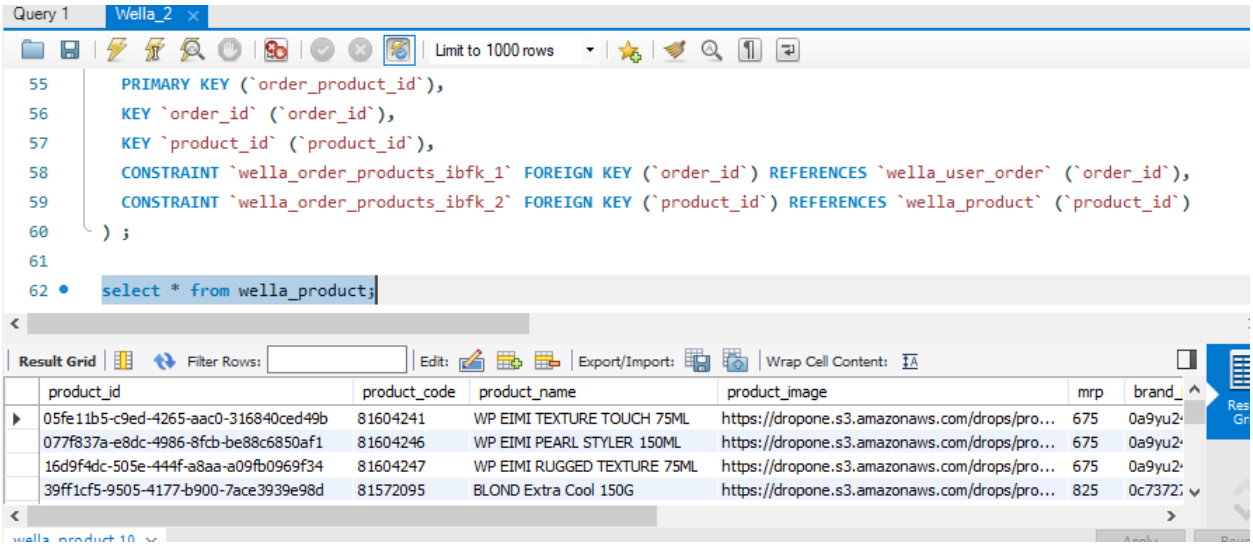
Create table for wella_order_products:

```
CREATE TABLE `wella_order_products` (
  `order_product_id` varchar(64) NOT NULL,
  `quantity` int DEFAULT NULL,
  `price` double DEFAULT NULL,
  `discount_price` double DEFAULT NULL,
  `bill_price` double DEFAULT NULL,
  `order_id` varchar(64) DEFAULT NULL,
  `product_id` varchar(64) DEFAULT NULL,
  `created_at` timestamp NOT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`order_product_id`),
  KEY `order_id` (`order_id`),
  KEY `product_id` (`product_id`),
  CONSTRAINT `wella_order_products_ibfk_1` FOREIGN KEY (`order_id`) REFERENCES `wella_user_order` (`order_id`),
```

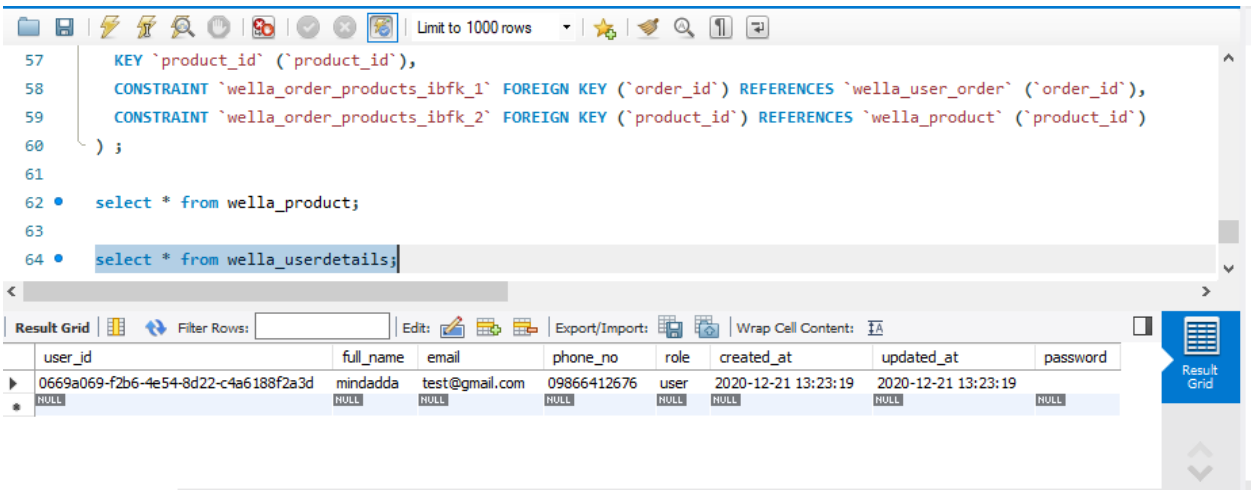
CONSTRAINT `wella_order_products_ibfk_2` FOREIGN KEY (`product_id`) REFERENCES `wella_product` (`product_id`)

) ;

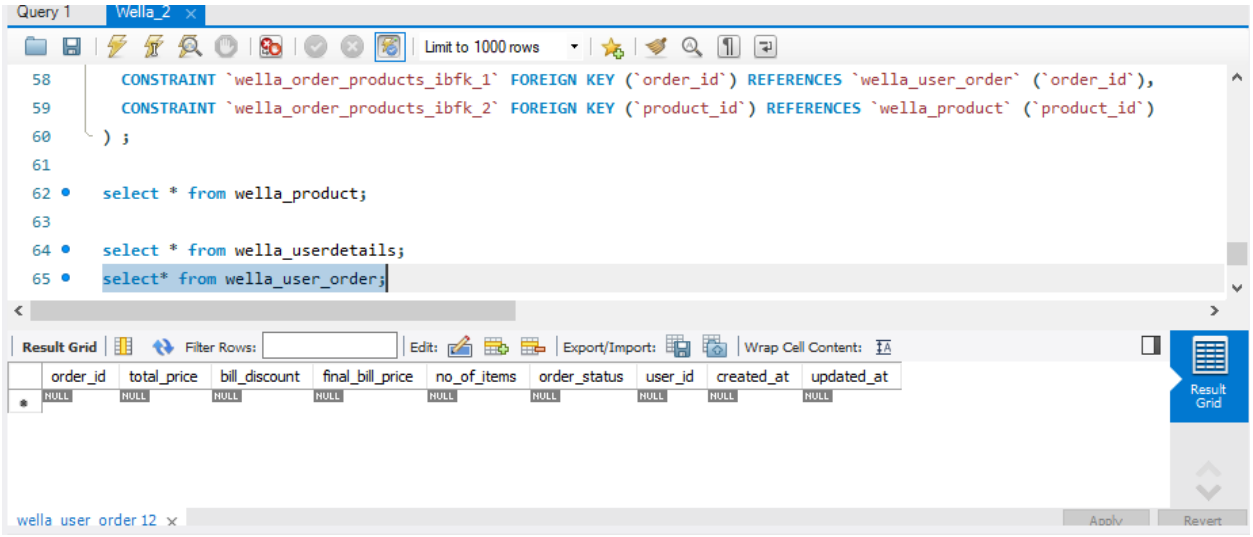
select * from wella_product;



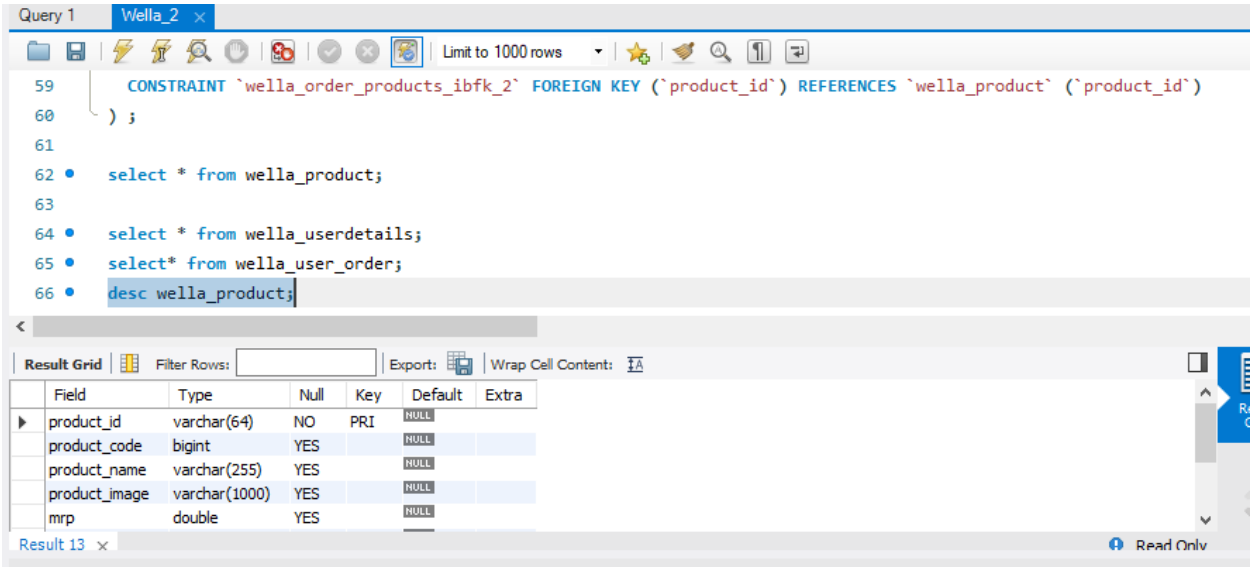
select * from wella_userdetails;



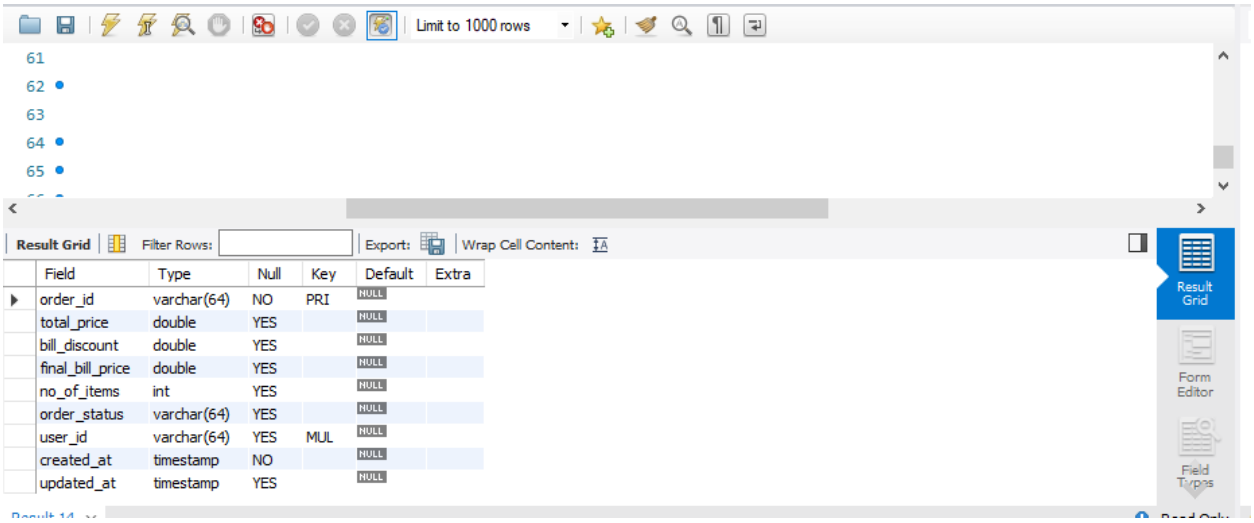
select* from wella_user_order;



desc wella_product;



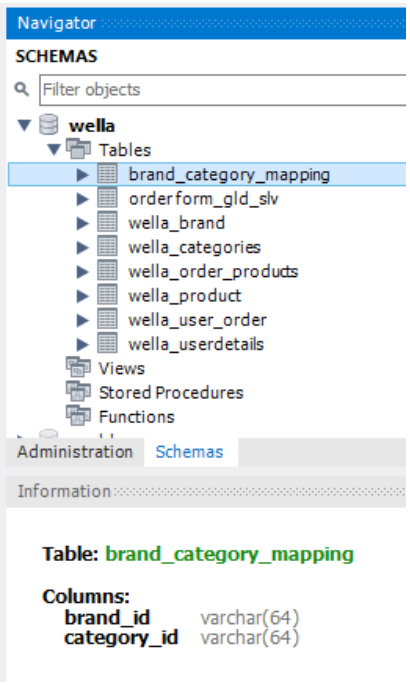
desc wella_user_order;

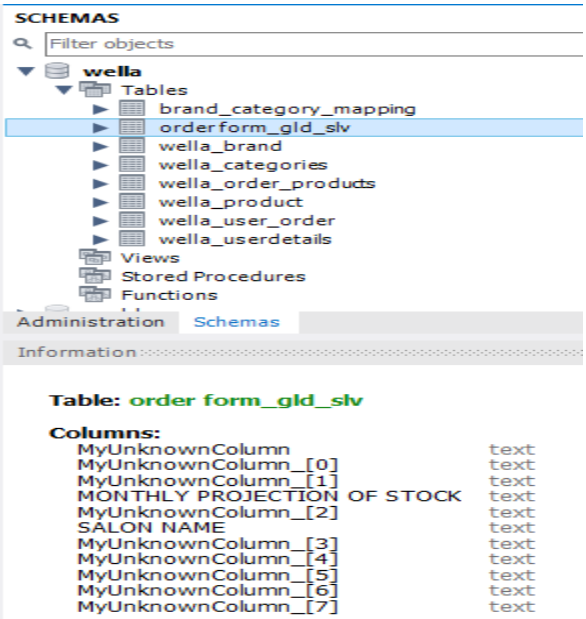


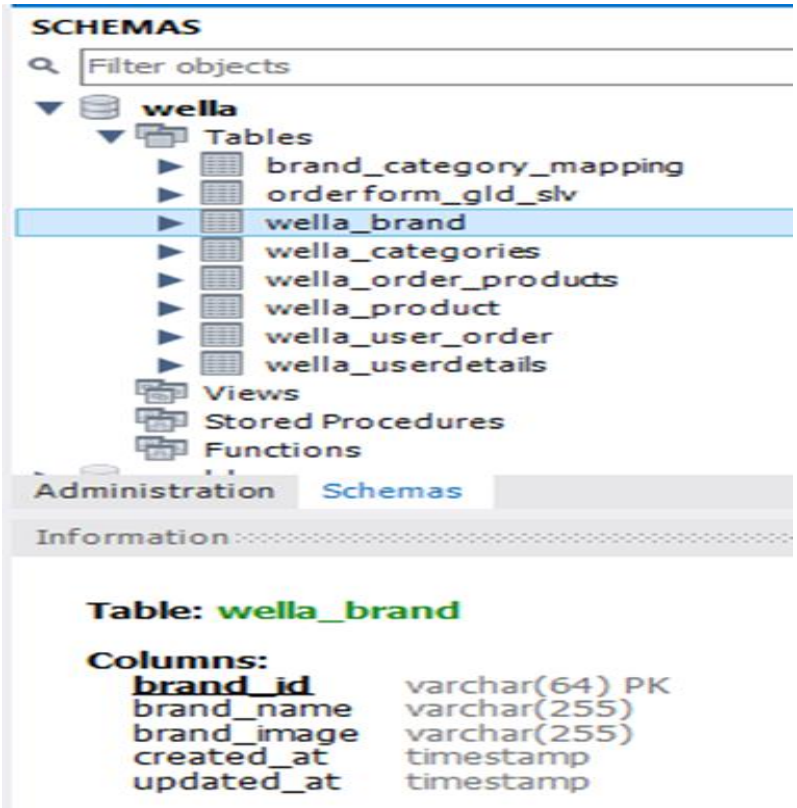
Field	Type	Null	Key	Default	Extra
order_id	varchar(64)	NO	PRI	NULL	
total_price	double	YES		NULL	
bill_discount	double	YES		NULL	
final_bill_price	double	YES		NULL	
no_of_items	int	YES		NULL	
order_status	varchar(64)	YES		NULL	
user_id	varchar(64)	YES	MUL	NULL	
created_at	timestamp	NO		NULL	
updated_at	timestamp	YES		NULL	

After creating all the tables , you can go to Schemas and check under ''wella'' , all the tables are present as follows :

And if you click a particular table then it will show the particular table name with the column details.


Navigator
SCHEMAS
Filter objects
wella
Tables
brand_category_mapping
orderform_gld_slv
wella_brand
wella_categories
wella_order_products
wella_product
wella_user_order
wella_userdetails
Views
Stored Procedures
Functions
Administration
Schemas
Information
Table: brand_category_mapping
Columns:
brand_id varchar(64)
category_id varchar(64)


SCHEMAS
Filter objects
wella
Tables
brand_category_mapping
orderform_gld_slv
wella_brand
wella_categories
wella_order_products
wella_product
wella_user_order
wella_userdetails
Views
Stored Procedures
Functions
Administration
Schemas
Information
Table: order form_gld_slv
Columns:
MyUnknownColumn text
MyUnknownColumn_[0] text
MyUnknownColumn_[1] text
MONTHLY PROJECTION OF STOCK text
MyUnknownColumn_[2] text
SALON NAME text
MyUnknownColumn_[3] text
MyUnknownColumn_[4] text
MyUnknownColumn_[5] text
MyUnknownColumn_[6] text
MyUnknownColumn_[7] text


SCHEMAS
Filter objects
wella
Tables
brand_category_mapping
orderform_gld_slv
wella_brand
wella_categories
wella_order_products
wella_product
wella_user_order
wella_userdetails
Views
Stored Procedures
Functions
Administration
Schemas
Information
Table: wella_brand
Columns:
brand_id varchar(64) PK
brand_name varchar(255)
brand_image varchar(255)
created_at timestamp
updated_at timestamp

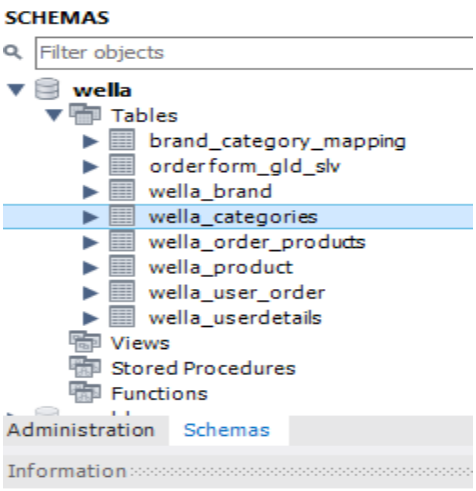
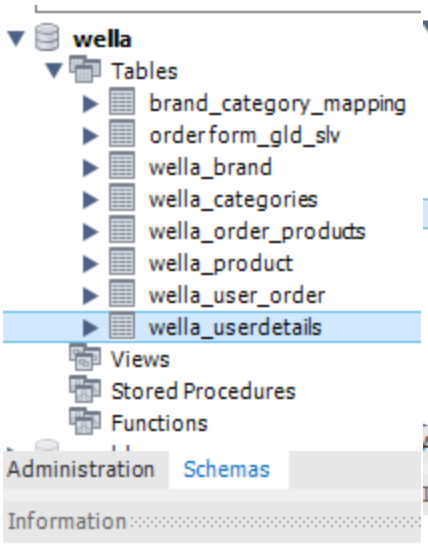


Table: wella_categories

Columns:

category_id	varchar(64) PK
category_name	varchar(255)
category_image	varchar(255)
brand_id	varchar(64)
created_at	timestamp
updated_at	timestamp



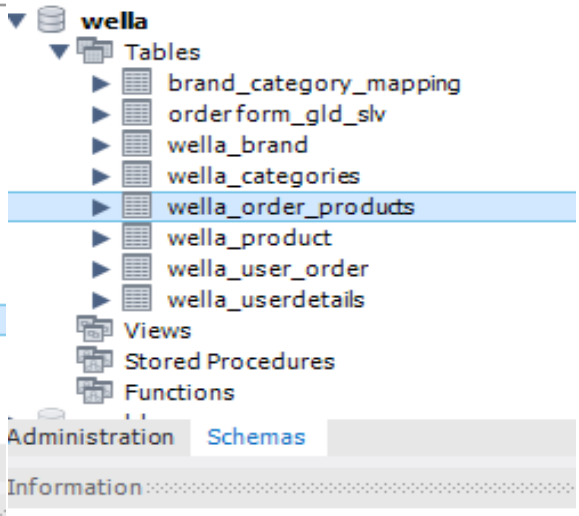


Table: wella_userdetails

Columns:

user_id	varchar(64) PK
full_name	varchar(255)
email	varchar(255)
phone_no	varchar(15)
role	varchar(10)
created_at	timestamp
updated_at	timestamp
password	varchar(255)

Table: wella_order_products

Columns:

order_product_id	varchar(64) PK
quantity	int
price	double
discount_price	double
bill_price	double
order_id	varchar(64)
product_id	varchar(64)
created_at	timestamp
updated_at	timestamp

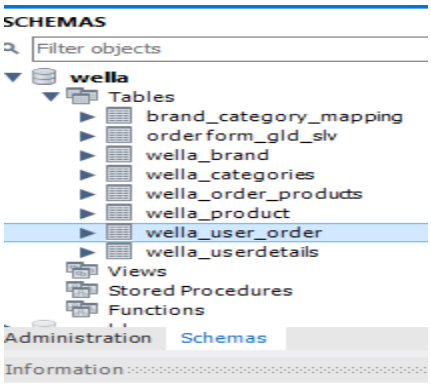


Table: wella_user_order

Columns:

order_id	varchar(64) PK
total_price	double
bill_discount	double
final_bill_price	double
no_of_items	int
order_status	varchar(64)
user_id	varchar(64)
created_at	timestamp
updated_at	timestamp

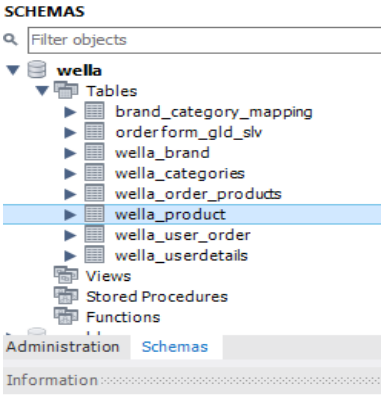


Table: wella_product

Columns:

product_id	varchar(64) PK
product_code	bigint
product_name	varchar(255)
product_image	varchar(1000)
mrp	double
brand_id	varchar(64)
category_id	varchar(64)
created_at	timestamp
updated_at	timestamp

Installing the Required Files

In [ ]:

```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
```

In [ ]:

```
!wget -q https://www-us.apache.org/dist/spark/spark-3.0.3/spark-3.0.3-bin-hadoop2.7.tgz
```

In [ ]:

```
!tar xf spark-3.0.3-bin-hadoop2.7.tgz
```

In [ ]:

```
!pip install -q findspark
```

In [ ]:

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.3-bin-hadoop2.7"
```

In [ ]:

```
import findspark
findspark.init()
```

In [ ]:

```
findspark.find()
```

Out[ ]:

```
'/content/spark-3.0.3-bin-hadoop2.7'
```

Launching Spark Session

In [ ]:

```
from pyspark.sql import SparkSession
# Launching Spark Session with Hive
spark = SparkSession.builder\
        .master("local")\
        .appName("Colab")\
        .config('spark.ui.port', '4050')\
        .enableHiveSupport()\
        .getOrCreate()
```

In [ ]:

```
spark
```

Out[ ]:

SparkSession - hive
SparkContext

Spark UI
Version
`v3.0.3`
Master
`local`

local

AppName

```
Colab
```

Optional

In [ ]:

```
!wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
!unzip ngrok-stable-linux-amd64.zip
get_ipython().system_raw('./ngrok http 4050 &')
!curl -s http://localhost:4040/api/tunnels
```

```
--2021-07-13 09:59:06--  https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
Resolving bin.equinox.io (bin.equinox.io)... 3.219.150.79, 3.223.73.198, 52.200.34.95, ...
Connecting to bin.equinox.io (bin.equinox.io)|3.219.150.79|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13832437 (13M) [application/octet-stream]
Saving to: 'ngrok-stable-linux-amd64.zip.1'

ngrok-stable-linux- 100%[===================>]  13.19M  12.7MB/s    in 1.0s

2021-07-13 09:59:08 (12.7 MB/s) - 'ngrok-stable-linux-amd64.zip.1' saved [13832437/13832437]

Archive:  ngrok-stable-linux-amd64.zip
replace ngrok? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  inflating: ngrok
{"tunnels":[{"name":"command_line (http)","uri":"/api/tunnels/command_line%20%28http%29","public_url":"http://5fded13889f5.ngrok.io","proto":"http","config":{"addr":"http://localhost:4050","inspect":true},"metrics":{"conns":{"count":0,"gauge":0,"rate1":0,"rate5":0,"rate15":0,"p50":0,"p90":0,"p95":0,"p99":0},"http":{"count":0,"rate1":0,"rate5":0,"rate15":0,"p50":0,"p90":0,"p95":0,"p99":0}}},{"name":"command_line","uri":"/api/tunnels/command_line","public_url":"https://5fded13889f5.ngrok.io","proto":"https","config":{"addr":"http://localhost:4050","inspect":true},"metrics":{"conns":{"count":0,"gauge":0,"rate1":0,"rate5":0,"rate15":0,"p50":0,"p90":0,"p95":0,"p99":0},"http":{"count":0,"rate1":0,"rate5":0,"rate15":0,"p50":0,"p90":0,"p95":0,"p99":0}}}],"uri":"/api/tunnels"}
```

Creating Spark DataFrame

A DataFrame is a distributed collection of data, which is organized into named columns. Conceptually, it is equivalent to relational tables with good optimization techniques.

A DataFrame can be constructed from an array of different sources such as Hive tables, Structured Data files, external databases, or existing RDDs. This API was designed for modern Big Data and data science applications taking inspiration from DataFrame in R Programming and Pandas in Python.

In [ ]:

```
df = spark.read.csv('/content/exchg_rate.csv',header=True,inferSchema=True)
```

In [ ]:

```
df1 = spark.read.csv('/content/exchg_rate.csv',header=False,inferSchema=True) # If you do not have the column names on your dataset
```

If you want to see the data in the DataFrame, then use the following command.

In [ ]:

```
df.show()
```

```
+---------+--------+---------------+--------------------+------------------+--------
```

```
------------+--------------------+--------------------+-------------+
|FROMCRNCCD|TOCRNCCD|CRNCFROMLONGDESC|      CRNCTOLONGDESC|CRNCEXCHGRATETYPECD|CRNCEXCHGR
ATETYPEDESC|CRNCEXCHGRATEEFFSTRTDT|CRNCEXCHGRATEEFFENDDT|CRNCEXCHGRATE|
+---------+-------+---------------+--------------------+-------------------+--------
------------+--------------------+--------------------+-------------+
|       SKK|     BDT| Slovakian Krona|     Bangladesh Taka|                  B|
excg rate|              01/11/05|             30/11/05|  3.182271075|
|       SKK|     USD| Slovakian Krona|     American Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.044820719|
|       SKK|   *UNK*| Slovakian Krona|                null|                  B|
excg rate|              01/11/05|             30/11/05|          0.0|
|       SKK|     IDR| Slovakian Krona|   Indonesian Rupiah|                  B|
excg rate|              01/11/05|             30/11/05|  403.3864743|
|       SKK|     MYR| Slovakian Krona|   Malaysian Ringgit|                  B|
excg rate|              01/11/05|             30/11/05|   0.13894423|
|       SKK|     PYG| Slovakian Krona|  Paraguayan Guarani|                  B|
excg rate|              01/11/05|             30/11/05|   210.657381|
|       SKK|     HRK| Slovakian Krona|       Croatian Kuna|                  B|
excg rate|              01/11/05|             30/11/05|  0.236909325|
|       SKK|     UYU| Slovakian Krona|Uruguayan Peso (new)|                  B|
excg rate|              01/11/05|             30/11/05|  0.896414387|
|       SKK|     KRW| Slovakian Krona|    South Korean Won|                  B|
excg rate|              01/11/05|             30/11/05|  51.54382727|
|       SKK|     ZWD| Slovakian Krona|   Zimbabwean Dollar|                  B|
excg rate|              01/11/05|             30/11/05|   0.16449204|
|       SKK|     CZK| Slovakian Krona|         Czech Krona|                  B|
excg rate|              01/11/05|             30/11/05|  0.800369861|
|       SKK|     THB| Slovakian Krona|       Thailand Baht|                  B|
excg rate|              01/11/05|             30/11/05|  1.344621581|
|       SKK|     AUD| Slovakian Krona|   Australian Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.047179705|
|       SKK|     VND| Slovakian Krona|     Vietnamese Dong|                  B|
excg rate|              01/11/05|             30/11/05|  941.2351066|
|       SKK|     HKD| Slovakian Krona|    Hong Kong Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.349601611|
|       SKK|     SGD| Slovakian Krona|    Singapore Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.056025899|
|       SKK|     CAD| Slovakian Krona|     Canadian Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.044820719|
|       SKK|     GBP| Slovakian Krona|       British Pound|                  B|
excg rate|              01/11/05|             30/11/05|   0.02721256|
|       SKK|     ZAR| Slovakian Krona|  South African Rand|                  B|
excg rate|              01/11/05|             30/11/05|  0.336155395|
|       SKK|     TRL| Slovakian Krona|        Turkish Lira|                  B|
excg rate|              01/11/05|             30/11/05|  71713.15098|
+---------+-------+---------------+--------------------+-------------------+--------
------------+--------------------+--------------------+-------------+
only showing top 20 rows
```

If you want to see the Structure (Schema) of the DataFrame, then use the following command.

In [ ]:

```
df.printSchema()
```

```
root
 |-- FROMCRNCCD: string (nullable = true)
 |-- TOCRNCCD: string (nullable = true)
 |-- CRNCFROMLONGDESC: string (nullable = true)
 |-- CRNCTOLONGDESC: string (nullable = true)
 |-- CRNCEXCHGRATETYPECD: string (nullable = true)
 |-- CRNCEXCHGRATETYPEDESC: string (nullable = true)
 |-- CRNCEXCHGRATEEFFSTRTDT: string (nullable = true)
 |-- CRNCEXCHGRATEEFFENDDT: string (nullable = true)
 |-- CRNCEXCHGRATE: double (nullable = true)
```

In [ ]:

```
df.count()
```

Out[ ]:

```
725671
```

If you want to see the miscalaneous statistics associated with the Spark Dataframe

In [ ]:

```
# Summary of Data
df.describe().show()
```

```
+-------+----------+--------+-----------------+-----------------+--------------------+--
------------------+----------------------+----------------------+------------------+
|summary|FROMCRNCCD|TOCRNCCD| CRNCFROMLONGDESC|   CRNCTOLONGDESC|CRNCEXCHGRATETYPECD|CRNC
EXCHGRATETYPEDESC|CRNCEXCHGRATEEFFSTRTDT|CRNCEXCHGRATEEFFENDDT|     CRNCEXCHGRATE|
+-------+----------+--------+-----------------+-----------------+--------------------+--
------------------+----------------------+----------------------+------------------+
|  count|    725671|  725671|           714598|           714598|              725671|
725671|                725671|                725671|            725671|
|   mean|      null|    null|             null|             null|                null|
null|                  null|                  null| 8617.573998703845|
| stddev|      null|    null|             null|             null|                null|
null|                  null|                  null|123838.61709580038|
|    min|     *UNK*|   *UNK*|  American Dollar|  American Dollar|                   B|
excg rate|            0001-01-01|              28/02/06|               0.0|
|    max|       ZWD|     ZWD|Zimbabwean Dollar|Zimbabwean Dollar|                   B|
excg rate|              01/12/21|              31/12/99|       4329510.879|
+-------+----------+--------+-----------------+-----------------+--------------------+--
------------------+----------------------+----------------------+------------------+
```

Some Basic Spark DataFrame Operations

In [ ]:

```
set1=df.select('FROMCRNCCD','TOCRNCCD','CRNCEXCHGRATE') # Creating a subset Dataframe
```

In [ ]:

```
set1.show()
```

```
+----------+--------+-------------+
|FROMCRNCCD|TOCRNCCD|CRNCEXCHGRATE|
+----------+--------+-------------+
|       SKK|     BDT|  3.182271075|
|       SKK|     USD|  0.044820719|
|       SKK|   *UNK*|          0.0|
|       SKK|     IDR|  403.3864743|
|       SKK|     MYR|   0.13894423|
|       SKK|     PYG|   210.657381|
|       SKK|     HRK|  0.236909325|
|       SKK|     UYU|  0.896414387|
|       SKK|     KRW|  51.54382727|
|       SKK|     ZWD|   0.16449204|
|       SKK|     CZK|  0.800369861|
|       SKK|     THB|  1.344621581|
|       SKK|     AUD|  0.047179705|
|       SKK|     VND|  941.2351066|
|       SKK|     HKD|  0.349601611|
|       SKK|     SGD|  0.056025899|
|       SKK|     CAD|  0.044820719|
|       SKK|     GBP|   0.02721256|
|       SKK|     ZAR|  0.336155395|
|       SKK|     TRL|  71713.15098|
+----------+--------+-------------+
only showing top 20 rows
```

Use the following command for finding the Currency which is Indian Rupee

Use the following command for finding the Currency which is Indian Rupee.

In [ ]:

```
df.filter(df.FROMCRNCCD=='INR').show()
```

```
+----------+--------+----------------+--------------------+------------------+--------
------------+----------------------+--------------------+--------------+
|FROMCRNCCD|TOCRNCCD|CRNCFROMLONGDESC|      CRNCTOLONGDESC|CRNCEXCHGRATETYPECD|CRNCEXCHGR
ATETYPEDESC|CRNCEXCHGRATEEFFSTRTDT|CRNCEXCHGRATEEFFENDDT|CRNCEXCHGRATE|
+----------+--------+----------------+--------------------+------------------+--------
------------+----------------------+--------------------+--------------+
|       INR|     BDT|    Indian Rupee|     Bangladesh Taka|                 B|
excg rate|              01/11/05|            30/11/05|  1.577777778|
|       INR|     USD|    Indian Rupee|     American Dollar|                 B|
excg rate|              01/11/05|            30/11/05|  0.022222222|
|       INR|   *UNK*|    Indian Rupee|                null|                 B|
excg rate|              01/11/05|            30/11/05|          0.0|
|       INR|     IDR|    Indian Rupee|   Indonesian Rupiah|                 B|
excg rate|              01/11/05|            30/11/05|        200.0|
|       INR|     MYR|    Indian Rupee|   Malaysian Ringgit|                 B|
excg rate|              01/11/05|            30/11/05|  0.068888889|
|       INR|     PYG|    Indian Rupee|  Paraguayan Guarani|                 B|
excg rate|              01/11/05|            30/11/05|  104.4444444|
|       INR|     HRK|    Indian Rupee|       Croatian Kuna|                 B|
excg rate|              01/11/05|            30/11/05|  0.117460222|
|       INR|     UYU|    Indian Rupee|Uruguayan Peso (new)|                 B|
excg rate|              01/11/05|            30/11/05|  0.444444444|
|       INR|     KRW|    Indian Rupee|    South Korean Won|                 B|
excg rate|              01/11/05|            30/11/05|  25.55555556|
|       INR|     ZWD|    Indian Rupee|   Zimbabwean Dollar|                 B|
excg rate|              01/11/05|            30/11/05|  0.081555556|
|       INR|     CZK|    Indian Rupee|         Czech Krona|                 B|
excg rate|              01/11/05|            30/11/05|  0.396825333|
|       INR|     THB|    Indian Rupee|       Thailand Baht|                 B|
excg rate|              01/11/05|            30/11/05|  0.666666667|
|       INR|     AUD|    Indian Rupee|   Australian Dollar|                 B|
excg rate|              01/11/05|            30/11/05|  0.023391813|
|       INR|     VND|    Indian Rupee|     Vietnamese Dong|                 B|
excg rate|              01/11/05|            30/11/05|  466.6666667|
|       INR|     HKD|    Indian Rupee|    Hong Kong Dollar|                 B|
excg rate|              01/11/05|            30/11/05|  0.173333333|
|       INR|     SGD|    Indian Rupee|    Singapore Dollar|                 B|
excg rate|              01/11/05|            30/11/05|  0.027777778|
|       INR|     CAD|    Indian Rupee|     Canadian Dollar|                 B|
excg rate|              01/11/05|            30/11/05|  0.022222222|
|       INR|     GBP|    Indian Rupee|       British Pound|                 B|
excg rate|              01/11/05|            30/11/05|  0.013492054|
|       INR|     ZAR|    Indian Rupee|  South African Rand|                 B|
excg rate|              01/11/05|            30/11/05|  0.166666667|
|       INR|     TRL|    Indian Rupee|        Turkish Lira|                 B|
excg rate|              01/11/05|            30/11/05|  35555.55556|
+----------+--------+----------------+--------------------+------------------+--------
------------+----------------------+--------------------+--------------+
only showing top 20 rows
```

Create Temporary View (HIVE Table)

In [ ]:

```
# Converting Spark Dataframe to Hive Table
df.write.mode("overwrite").saveAsTable("test_table2")
```

In [ ]:

```
#Instances where INR Present
INR=spark.sql("select * from test_table2  a where a.FROMCRNCCD=='INR'")
```

In [ ]:

```
INR.show()
```

```
+----------+--------+----------------+--------------------+-------------------+--------
------------+----------------------+--------------------+-------------+
|FROMCRNCCD|TOCRNCCD|CRNCFROMLONGDESC|      CRNCTOLONGDESC|CRNCEXCHGRATETYPECD|CRNCEXCHGR
ATETYPEDESC|CRNCEXCHGRATEEFFSTRTDT|CRNCEXCHGRATEEFFENDDT|CRNCEXCHGRATE|
+----------+--------+----------------+--------------------+-------------------+--------
------------+----------------------+--------------------+-------------+
|       INR|     BDT|    Indian Rupee|     Bangladesh Taka|                  B|
excg rate|              01/11/05|             30/11/05|  1.577777778|
|       INR|     USD|    Indian Rupee|     American Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.022222222|
|       INR|   *UNK*|    Indian Rupee|                null|                  B|
excg rate|              01/11/05|             30/11/05|          0.0|
|       INR|     IDR|    Indian Rupee|   Indonesian Rupiah|                  B|
excg rate|              01/11/05|             30/11/05|        200.0|
|       INR|     MYR|    Indian Rupee|   Malaysian Ringgit|                  B|
excg rate|              01/11/05|             30/11/05|  0.068888889|
|       INR|     PYG|    Indian Rupee|  Paraguayan Guarani|                  B|
excg rate|              01/11/05|             30/11/05|  104.4444444|
|       INR|     HRK|    Indian Rupee|       Croatian Kuna|                  B|
excg rate|              01/11/05|             30/11/05|  0.117460222|
|       INR|     UYU|    Indian Rupee|Uruguayan Peso (new)|                  B|
excg rate|              01/11/05|             30/11/05|  0.444444444|
|       INR|     KRW|    Indian Rupee|    South Korean Won|                  B|
excg rate|              01/11/05|             30/11/05|  25.55555556|
|       INR|     ZWD|    Indian Rupee|   Zimbabwean Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.081555556|
|       INR|     CZK|    Indian Rupee|         Czech Krona|                  B|
excg rate|              01/11/05|             30/11/05|  0.396825333|
|       INR|     THB|    Indian Rupee|       Thailand Baht|                  B|
excg rate|              01/11/05|             30/11/05|  0.666666667|
|       INR|     AUD|    Indian Rupee|   Australian Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.023391813|
|       INR|     VND|    Indian Rupee|     Vietnamese Dong|                  B|
excg rate|              01/11/05|             30/11/05|  466.6666667|
|       INR|     HKD|    Indian Rupee|    Hong Kong Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.173333333|
|       INR|     SGD|    Indian Rupee|    Singapore Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.027777778|
|       INR|     CAD|    Indian Rupee|     Canadian Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.022222222|
|       INR|     GBP|    Indian Rupee|       British Pound|                  B|
excg rate|              01/11/05|             30/11/05|  0.013492054|
|       INR|     ZAR|    Indian Rupee|  South African Rand|                  B|
excg rate|              01/11/05|             30/11/05|  0.166666667|
|       INR|     TRL|    Indian Rupee|        Turkish Lira|                  B|
excg rate|              01/11/05|             30/11/05|  35555.55556|
+----------+--------+----------------+--------------------+-------------------+--------
------------+----------------------+--------------------+-------------+
only showing top 20 rows
```

In [ ]:

```
#Instances where INR to Euro Present
INR_EUR=spark.sql("select * from test_table2  where test_table2.FROMCRNCCD=='INR' AND tes
t_table2.TOCRNCCD=='EUR'")
```

In [ ]:

```
INR_EUR.show()
```

```
+----------+--------+----------------+--------------------+-------------------+---------
-----------+----------------------+--------------------+-------------+
|FROMCRNCCD|TOCRNCCD|CRNCFROMLONGDESC|      CRNCTOLONGDESC|CRNCEXCHGRATETYPECD|CRNCEXCHGRA
TETYPEDESC|CRNCEXCHGRATEEFFSTRTDT|CRNCEXCHGRATEEFFENDDT|CRNCEXCHGRATE|
+----------+--------+----------------+--------------------+-------------------+---------
-----------+----------------------+--------------------+-------------+
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/11/05|             30/11/05|  0.015873016|
```

```
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/12/05|             31/12/05|  0.015873016|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/01/06|             31/01/06|  0.015873016|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/02/06|             28/02/06|  0.015873016|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/03/06|             31/03/06|  0.015873016|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/04/06|             30/04/06|  0.015873016|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/05/06|             31/05/06|  0.015873016|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/06/06|             30/06/06|  0.015873016|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/07/06|             31/07/06|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/08/06|             31/08/06|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/09/06|             30/09/06|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/10/06|             31/10/06|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/11/06|             30/11/06|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/12/06|             31/12/06|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/01/07|             31/01/07|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/02/07|             28/02/07|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/03/07|             31/03/07|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/04/07|             30/04/07|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/05/07|             31/05/07|  0.015384615|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                  B|
excg rate|              01/06/07|             30/06/07|  0.015384615|
+----------+--------+----------------+-------------------+-------------------+---------
-----------+---------------------+---------------------+-------------+
only showing top 20 rows
```

In [ ]:

```
#Instances where INR to USD Present
INR_USD=spark.sql("select * from test_table2  a where a.FROMCRNCCD=='INR' AND a.TOCRNCCD
=='USD'")
```

In [ ]:

```
INR_USD.show()
```

```
+----------+--------+----------------+---------------+-------------------+-------------
--------+---------------------+---------------------+-------------+
|FROMCRNCCD|TOCRNCCD|CRNCFROMLONGDESC| CRNCTOLONGDESC|CRNCEXCHGRATETYPECD|CRNCEXCHGRATETY
PEDESC|CRNCEXCHGRATEEFFSTRTDT|CRNCEXCHGRATEEFFENDDT|CRNCEXCHGRATE|
+----------+--------+----------------+---------------+-------------------+-------------
--------+---------------------+---------------------+-------------+
|       INR|     USD|    Indian Rupee|American Dollar|                  B|            ex
cg rate|              01/11/05|             30/11/05|  0.022222222|
|       INR|     USD|    Indian Rupee|American Dollar|                  B|            ex
cg rate|              01/12/05|             31/12/05|  0.022222222|
|       INR|     USD|    Indian Rupee|American Dollar|                  B|            ex
cg rate|              01/01/06|             31/01/06|  0.022222222|
|       INR|     USD|    Indian Rupee|American Dollar|                  B|            ex
cg rate|              01/02/06|             28/02/06|  0.022222222|
|       INR|     USD|    Indian Rupee|American Dollar|                  B|            ex
cg rate|              01/03/06|             31/03/06|  0.022222222|
|       INR|     USD|    Indian Rupee|American Dollar|                  B|            ex
cg rate|              01/04/06|             30/04/06|  0.022222222|
|       INR|     USD|    Indian Rupee|American Dollar|                  B|            ex
```

```
cg rate|              01/05/06|              31/05/06|  0.022222222|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/06/06|              30/06/06|  0.022222222|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/07/06|              31/07/06|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/08/06|              31/08/06|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/09/06|              30/09/06|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/10/06|              31/10/06|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/11/06|              30/11/06|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/12/06|              31/12/06|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/01/07|              31/01/07|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/02/07|              28/02/07|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/03/07|              31/03/07|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/04/07|              30/04/07|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/05/07|              31/05/07|  0.019230769|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/06/07|              30/06/07|  0.019230769|
+----------+--------+---------------+---------------+--------------------+-------------
--------+----------------------+--------------------+-------------+
only showing top 20 rows
```

Transformation Function and creating a new column in DataFrames

Here, the Table gives 1 INR= x USD If you want ti get 1 USD, by Unitary method it will be: 1 USD= 1/x USD

In [ ]:

```
# Converting INR to USD
INR_USD=INR_USD.withColumn("BACK", 1/INR_USD.CRNCEXCHGRATE)
```

In [ ]:

```
INR_USD.show()
```

```
+----------+--------+---------------+---------------+--------------------+-------------
--------+----------------------+--------------------+-------------+------------+
|FROMCRNCCD|TOCRNCCD|CRNCFROMLONGDESC| CRNCTOLONGDESC|CRNCEXCHGRATETYPECD|CRNCEXCHGRATETY
PEDESC|CRNCEXCHGRATEEFFSTRTDT|CRNCEXCHGRATEEFFENDDT|CRNCEXCHGRATE|        BACK|
+----------+--------+---------------+---------------+--------------------+-------------
--------+----------------------+--------------------+-------------+------------+
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/11/05|              30/11/05|  0.022222222| 45.00000045|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/12/05|              31/12/05|  0.022222222| 45.00000045|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/01/06|              31/01/06|  0.022222222| 45.00000045|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/02/06|              28/02/06|  0.022222222| 45.00000045|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/03/06|              31/03/06|  0.022222222| 45.00000045|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/04/06|              30/04/06|  0.022222222| 45.00000045|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/05/06|              31/05/06|  0.022222222| 45.00000045|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|              01/06/06|              30/06/06|  0.022222222| 45.00000045|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
```

```
cg rate|               01/07/06|             31/07/06|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/08/06|             31/08/06|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/09/06|             30/09/06|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/10/06|             31/10/06|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/11/06|             30/11/06|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/12/06|             31/12/06|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/01/07|             31/01/07|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/02/07|             28/02/07|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/03/07|             31/03/07|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/04/07|             30/04/07|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/05/07|             31/05/07|  0.019230769|52.000000624|
|       INR|     USD|    Indian Rupee|American Dollar|                    B|            ex
cg rate|               01/06/07|             30/06/07|  0.019230769|52.000000624|
+----------+--------+----------------+---------------+--------------------+-------------
--------+----------------------+--------------------+-------------+------------+
only showing top 20 rows
```

In [ ]:

```
# Converting INR to Euro
INR_EUR=INR_EUR.withColumn("BACK", 1/INR_USD.CRNCEXCHGRATE)
```

In [ ]:

```
INR_EUR.show()
```

```
+----------+--------+----------------+--------------------+--------------------+---------
-----------+----------------------+--------------------+-------------+---------------
--+
|FROMCRNCCD|TOCRNCCD|CRNCFROMLONGDESC|     CRNCTOLONGDESC|CRNCEXCHGRATETYPECD|CRNCEXCHGRA
TETYPEDESC|CRNCEXCHGRATEEFFSTRTDT|CRNCEXCHGRATEEFFENDDT|CRNCEXCHGRATE|           BACK|
+----------+--------+----------------+--------------------+--------------------+---------
-----------+----------------------+--------------------+-------------+---------------
--+
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/11/05|             30/11/05|  0.015873016|     62.999999496|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/12/05|             31/12/05|  0.015873016|     62.999999496|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/01/06|             31/01/06|  0.015873016|     62.999999496|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/02/06|             28/02/06|  0.015873016|     62.999999496|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/03/06|             31/03/06|  0.015873016|     62.999999496|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/04/06|             30/04/06|  0.015873016|     62.999999496|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/05/06|             31/05/06|  0.015873016|     62.999999496|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/06/06|             30/06/06|  0.015873016|     62.999999496|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/07/06|             31/07/06|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/08/06|             31/08/06|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/09/06|             30/09/06|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/10/06|             31/10/06|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|               01/11/06|             30/11/06|  0.015384615|65.00000162500004|
```

```
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|              01/12/06|             31/12/06|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|              01/01/07|             31/01/07|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|              01/02/07|             28/02/07|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|              01/03/07|             31/03/07|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|              01/04/07|             30/04/07|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|              01/05/07|             31/05/07|  0.015384615|65.00000162500004|
|       INR|     EUR|    Indian Rupee|EMU currency (Euro)|                   B|
excg rate|              01/06/07|             30/06/07|  0.015384615|65.00000162500004|
+----------+--------+---------------+-------------------+------------------+---------
-----------+----------------------+---------------------+-------------+---------------
--+
only showing top 20 rows
```

In [ ]:

```
# If we are doing this to whole DataFrame
df=df.withColumn("BACK", 1/df.CRNCEXCHGRATE)
```

In [ ]:

```
df.show()
```

```
+----------+--------+---------------+-------------------+------------------+--------
-------------+----------------------+---------------------+-------------+--------------
------+
|FROMCRNCCD|TOCRNCCD|CRNCFROMLONGDESC|      CRNCTOLONGDESC|CRNCEXCHGRATETYPECD|CRNCEXCHGR
ATETYPEDESC|CRNCEXCHGRATEEFFSTRTDT|CRNCEXCHGRATEEFFENDDT|CRNCEXCHGRATE|                BA
CK|
+----------+--------+---------------+-------------------+------------------+--------
-------------+----------------------+---------------------+-------------+--------------
------+
|       SKK|     BDT| Slovakian Krona|     Bangladesh Taka|                 B|
excg rate|              01/11/05|             30/11/05|  3.182271075|  0.314240985897155
2|
|       SKK|     USD| Slovakian Krona|     American Dollar|                 B|
excg rate|              01/11/05|             30/11/05|  0.044820719|  22.31111018098571
8|
|       SKK|   *UNK*| Slovakian Krona|               null|                 B|
excg rate|              01/11/05|             30/11/05|          0.0|               nul
l|
|       SKK|     IDR| Slovakian Krona|   Indonesian Rupiah|                 B|
excg rate|              01/11/05|             30/11/05|  403.3864743|0.00247901222205159
1|
|       SKK|     MYR| Slovakian Krona|   Malaysian Ringgit|                 B|
excg rate|              01/11/05|             30/11/05|   0.13894423|   7.19713225946842
1|
|       SKK|     PYG| Slovakian Krona|  Paraguayan Guarani|                 B|
excg rate|              01/11/05|             30/11/05|   210.657381|0.00474704468105012
7|
|       SKK|     HRK| Slovakian Krona|       Croatian Kuna|                 B|
excg rate|              01/11/05|             30/11/05|  0.236909325|   4.22102422519670
7|
|       SKK|     UYU| Slovakian Krona|Uruguayan Peso (new)|                 B|
excg rate|              01/11/05|             30/11/05|  0.896414387|  1.115555500338037
3|
|       SKK|     KRW| Slovakian Krona|    South Korean Won|                 B|
excg rate|              01/11/05|             30/11/05|  51.54382727|0.01940096521668325
7|
|       SKK|     ZWD| Slovakian Krona|   Zimbabwean Dollar|                 B|
excg rate|              01/11/05|             30/11/05|   0.16449204|   6.07932152826361
7|
|       SKK|     CZK| Slovakian Krona|         Czech Krona|                 B|
excg rate|              01/11/05|             30/11/05|  0.800369861|  1.249422359245983
7|
|       SKK|     THB| Slovakian Krona|       Thailand Baht|                 B|
```

```
excg rate|              01/11/05|             30/11/05|  1.344621581|  0.743703666615477
3|
|       SKK|     AUD| Slovakian Krona|   Australian Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.047179705|  21.19555431726417
2|
|       SKK|     VND| Slovakian Krona|     Vietnamese Dong|                  B|
excg rate|              01/11/05|             30/11/05|  941.2351066|0.001062433809563..
.|
|       SKK|     HKD| Slovakian Krona|    Hong Kong Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.349601611|   2.86039871824274
9|
|       SKK|     SGD| Slovakian Krona|    Singapore Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.056025899|  17.84888806514287
3|
|       SKK|     CAD| Slovakian Krona|     Canadian Dollar|                  B|
excg rate|              01/11/05|             30/11/05|  0.044820719|  22.31111018098571
8|
|       SKK|     GBP| Slovakian Krona|       British Pound|                  B|
excg rate|              01/11/05|             30/11/05|   0.02721256|   36.7477370743509
6|
|       SKK|     ZAR| Slovakian Krona|  South African Rand|                  B|
excg rate|              01/11/05|             30/11/05|  0.336155395|  2.974814668674289
6|
|       SKK|     TRL| Slovakian Krona|        Turkish Lira|                  B|
excg rate|              01/11/05|             30/11/05|  71713.15098|1.394444375033651..
.|
+----------+--------+----------------+--------------------+------------------+--------
------------+----------------------+--------------------+-------------+-------------
------+
only showing top 20 rows
```

In [ ]:

```
from pyspark.sql.functions import col, avg
# Average of Euro Currency
INR_EUR.agg(avg(col("BACK"))).show()
# Average of USD Currency
INR_USD.agg(avg(col("BACK"))).show()
```

```
+-----------------+
|        avg(BACK)|
+-----------------+
|79.23437279809256|
+-----------------+

+----------------+
|       avg(BACK)|
+----------------+
|66.7437176038998|
+----------------+
```

Naïve Bayes Model and Directed Graphical Model

Akash Choudhuri
Roll: 2019D014

M.Sc (2nd Year), Mathematics with Data Science
Institute of Mathematics & Applications, Bhubaneswar
akashchoudhuri.ima@iomaorissa.ac.in

July, 2021

Outline

- Conditional Independence and Bayes Theorem.
- The Naïve Bayes Model.
- Directed Graphical Models.
- Bayesian Networks.
- Application
- Programmed Example (if time permits).

Conditional Independence and Bayes Theorem

Axioms of Probability Theorem:

- For an event A, the probability of occurrence of that event A will be greater than or equal to zero.

$$p(A) \geq 0$$

- If there are disjoint events in a sample space, then the union of all events is the summation of individual probabilities.

$$P\left(\bigcup A_i\right) = \sum_i P(A_i)$$

- In case of an event involving the universal set has the probability of 1.

Important Concepts of Probability Theory

- **Random Variable:** A random variable is a measurable function which maps each outcome of the sample space to a Real value.
- **Joint Probability Distribution:** It finds the probability of many events occurring together by treating each event as a random variable. Eg, for 3 events X_1, X_2, X_3, Joint distribution is denoted by $P(X_1, X_2, X_3)$.
- **Marginal Probability Distribution:** Let X_1, X_2, X_3 be 3 random variables. Then the marginal distribution is:

$$P(x_1) = \sum \Sigma^p (x_1, x_2, x_3)$$

Introduction to Bayes Theorem

- **Conditional Independence:** We say an event X is conditionally independent of event Y given an event Z denoted as:

$$P(X|Y, Z) = P(X|Z).$$

- **Bayes Theorem:** Principled way of calculating a conditional probability without the joint probability.

In simpler terms, the result P(A|B) is referred to as the posterior probability and P(A) is referred to as the prior probability. Sometimes P(B|A) is referred to as the likelihood and P(B) is referred to as the evidence. This allows Bayes Theorem to be restated as:

$$\text{Posterior} = \text{Likelihood} * \text{Prior} / \text{Evidence}$$

The Naïve Bayes Model

Why ‘naïve’?

- This model uses Bayes Theorem with a small assumption that **there is independence among predictors**, ie, the presence of a particular feature in a class is unrelated to the presence of any other feature.

So, our Bayes Theorem formula is re-written by omitting the denominator (a littler bit of maths can show that and it reduces to:

By Bayes Theorem, $P(B|A)=(P(A|B)*P(B))/P(A)$

$=P(A|B)*P(B)$

Generalising the Equation,

$$P(c|X)=P(x_1|c)*P(x_2|c)*P(x_3|c)\ldots\ldots P(x_n|c)*P(c)$$

Naïve Bayes Classifier Algorithm for Discrete Data

- **Step 1:** Given a set of features D containing target variable T, calculate $P(X_i \mid Y_i)$ where

$$X_i, Y_i \in D \text{ and } X_i \neq Y_i$$

- **Step 2:** Calculate the Class Probabilities of Y given as P(Y).
- **Step 3:** Train the Model by finding the probabilities.
- **Step 4:** For a new set of features which is a subset of D, find the corresponding T.

Directed Graphical Models

Kinds of Graphical Models

- Undirected Graphical Models also known as Markov Random Fields.
- Directed graphical models also known as Bayesian (belief) networks. The important Characteristics of Bayesian Networks are:
  - Bayesian Networks require that the graph is a DAG (directed acyclic graphs).
  - No directed cycles allowed.

Bayesian Networks

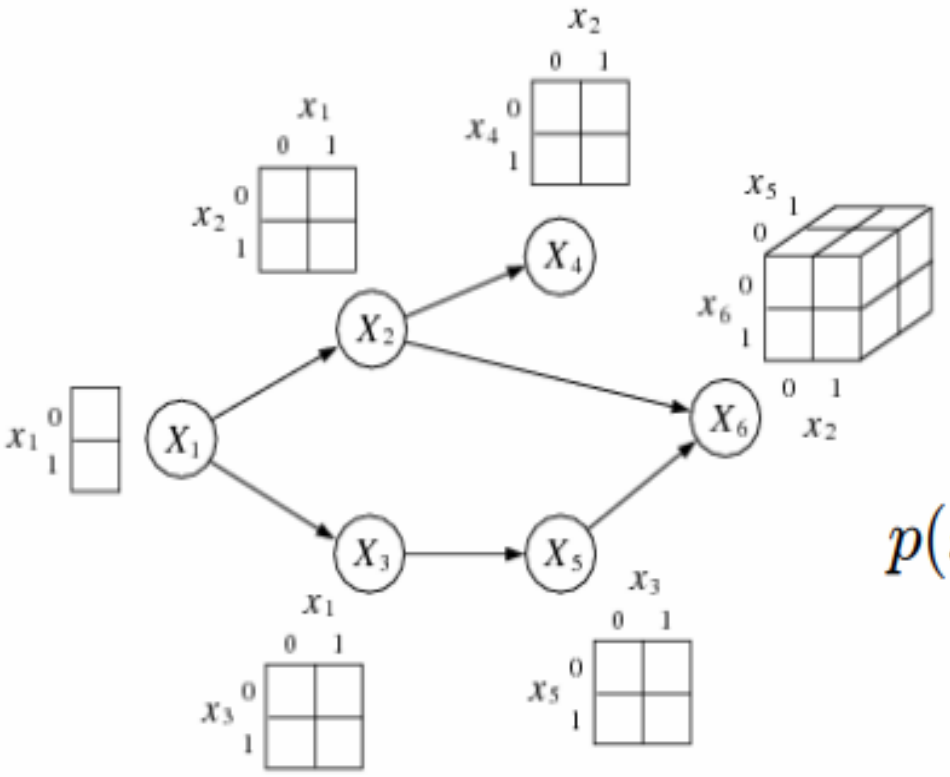
Bayesian Networks

- Judea Paul, who is credited with the invention of Bayesian Networks, won the Turing Award in 2011 for this discovery.
- A probability distribution factorizes according to a DAG if it can be written as:

$$P(x) = \prod_{j=1}^{d} P\left(x_j \middle| x_{\pi j}\right)$$

Where Π_j are the parents of j , and the nodes are ordered topologically (parents before children).

Continued….



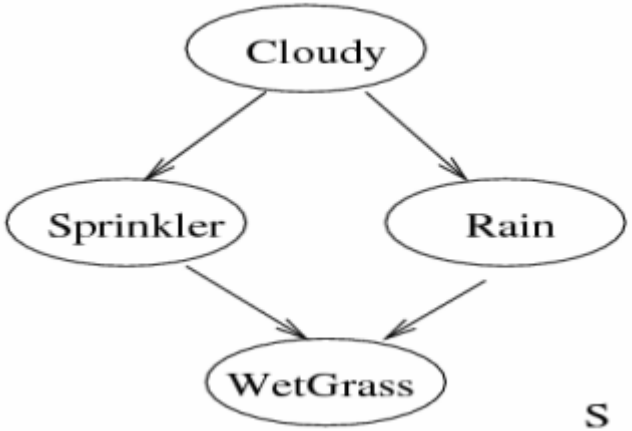
Each row of the conditional probability table (CPT) defines the distribution over the child's values given its parents values. The model is locally normalized.

$$p(x_{1:6}) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_3)$$
$$p(x_5|x_2,x_3)p(x_6|x_2,x_5)$$

Example Bayesian Network

P(C=F)	P(C=T)
0.5	0.5

C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1



C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

Continued

c	s	r	w	prob
0	0	0	0	0.200
0	0	0	1	0.000
0	0	1	0	0.005
0	0	1	1	0.045
0	1	0	0	0.020
0	1	0	1	0.180
0	1	1	0	0.001
0	1	1	1	0.050
1	0	0	0	0.090
1	0	0	1	0.000
1	0	1	0	0.036
1	0	1	1	0.324
1	1	0	0	0.001
1	1	0	1	0.009
1	1	1	0	0.000
1	1	1	1	0.040

- The joint distribution is computed using Naïve Bayes Model as:

$$p(C, S, R, W) = p(C)\, p(S|C)\, p(R|C)\, p(W|S, R)$$

- Prior that sprinkler is on:

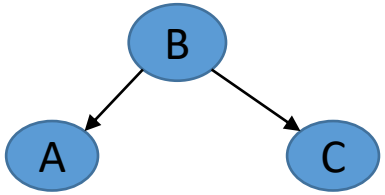
$$p(S=1) = \sum_{c=0}^{1} \sum_{r=0}^{1} \sum_{w=0}^{1} p(C=c, S=1, R=r, W=w) = 0.3$$

- Posterior that sprinkler is on given that grass is wet:

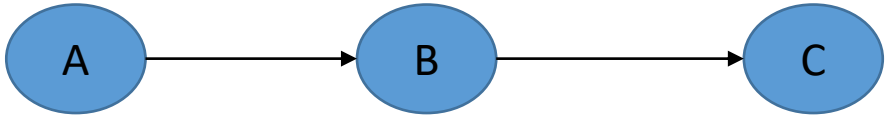
$$p(S=1|W=1) = \frac{p(S=1, W=1)}{p(W=1)} = 0.43$$

Conditional Independencies Implied from Bayesian Networks

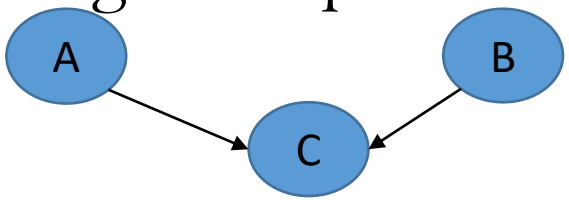
- **Common Parent:** Fixing B, A and C are decoupled in this network ($A \perp C \mid B$).



- **Cascade Structure:** In this network, $A \perp C \mid B$.



- **V- Structure:** Knowing C couples A & B.



D- Separation

Let A,B &C be non-overlapping sets of nodes (vertices) of a graph G. To ascertain $(A \perp B \mid C)$, consider all paths from any node in A to any node in B. Any such path is said to be block if it includes a node such that:

- The arrows on the path meet either head-to-tail or tail-to-tail and the node is in the set C.

OR

- The arrows meet head-to-head at the nodes and neither the node nor any of its descendants is in the set C.

Fact: If A is d-separated from B by C, then $(A \perp B \mid C)$ holds in the graph.

Application

A Bayesian Network Model for Predicting Post stroke Outcomes With Available Risk Factors

- An inference engine was constructed for post-stroke outcomes based on Bayesian network classifiers.
- The prediction system that was trained on data of 3,605 patients with acute stroke forecasts the functional independence at 3 months and the mortality 1 year after stroke.
- Feature selection methods were applied to eliminate less relevant and redundant features from 76 risk variables.
- Bayesian network with selected features by wrapper-type feature selection can predict 3-month functional independence with an AUC of 0.889 using only 19 risk variables and 1-year mortality with an AUC of 0.893 using 24 variables.

Dataset

- During admission, all patients were thoroughly investigated for medical history, clinical manifestations, and the presence of vascular risk factors.
- All registered patients underwent brain imaging studies including brain computed tomography (CT) and/or MRI.
- Stroke classification was determined during weekly conferences based on the consensus of stroke neurologists. Data including clinical information, risk factors, imaging study findings, laboratory analyses, and other special evaluations were collected. Along with these data, prognosis during hospitalization and long-term outcomes were also determined.

Methodology

- 76 Random variables were extracted from the data. Then a Bayesian Network was constructed using the formulae given before.
- Given a data set D with variable V_i, the observed distribution P_D is described as a joint probability distribution over D. The learning process now measures and compares the quality of Bayesian networks to evaluate how well the represented distribution explains the given data set. The log-likelihood is the basic common value used for measuring the quality of a Bayesian network as follows:

$$LL(B|D) = \sum_{V_i} log(P(V_i | \pi_B(V_i))),$$

Methodology cont.

- The algorithm searched the best Bayesian network based on the Bayesian information criterion. In this case maximum description length (MDL) score was used as evaluator. The MDL score is described as:

$$\mathrm{MDL} = -\mathrm{LL}(\mathrm{B}|\mathrm{D}) + \frac{\log N}{2} |B|$$

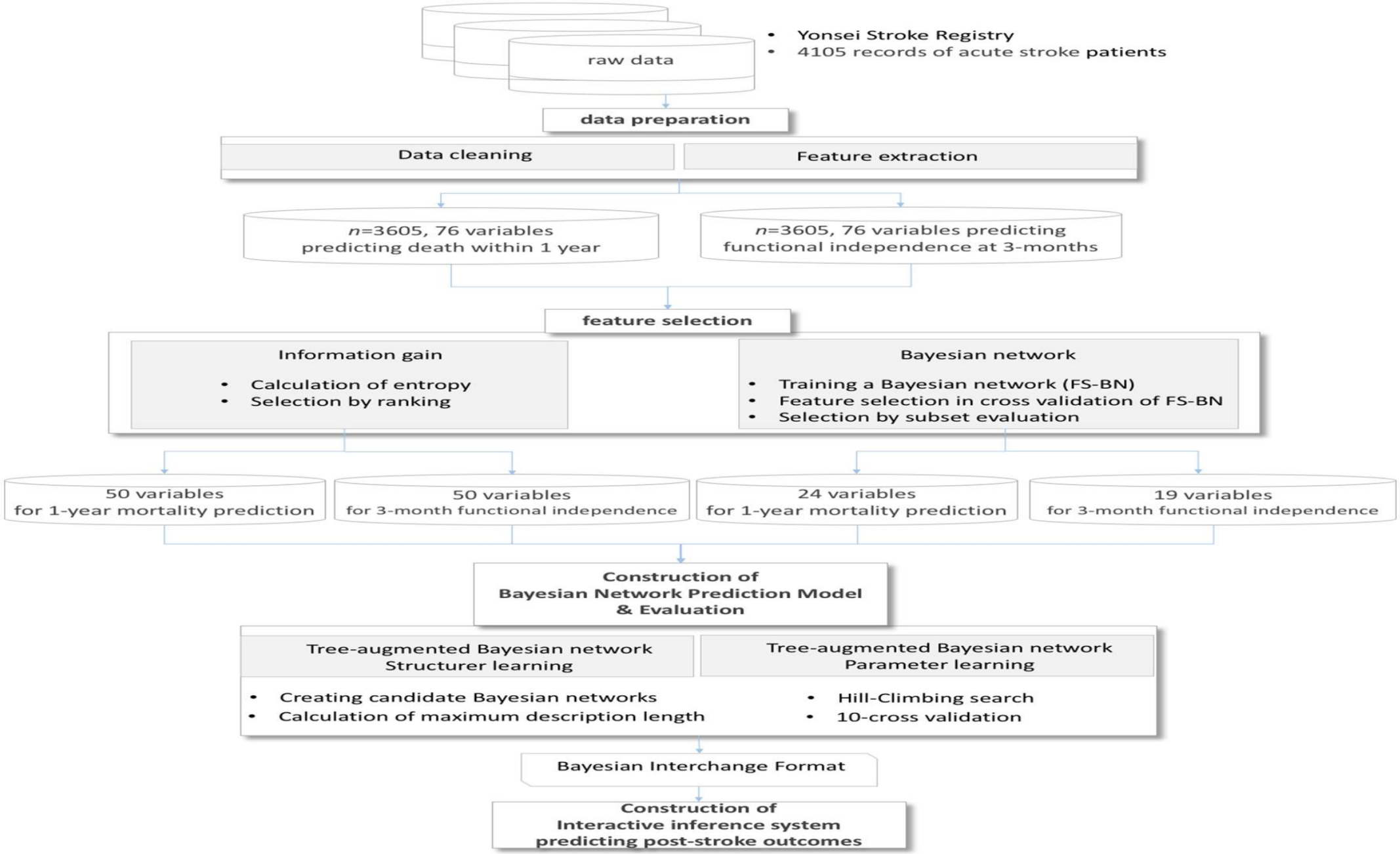
Where,

N is the number of instances in D,

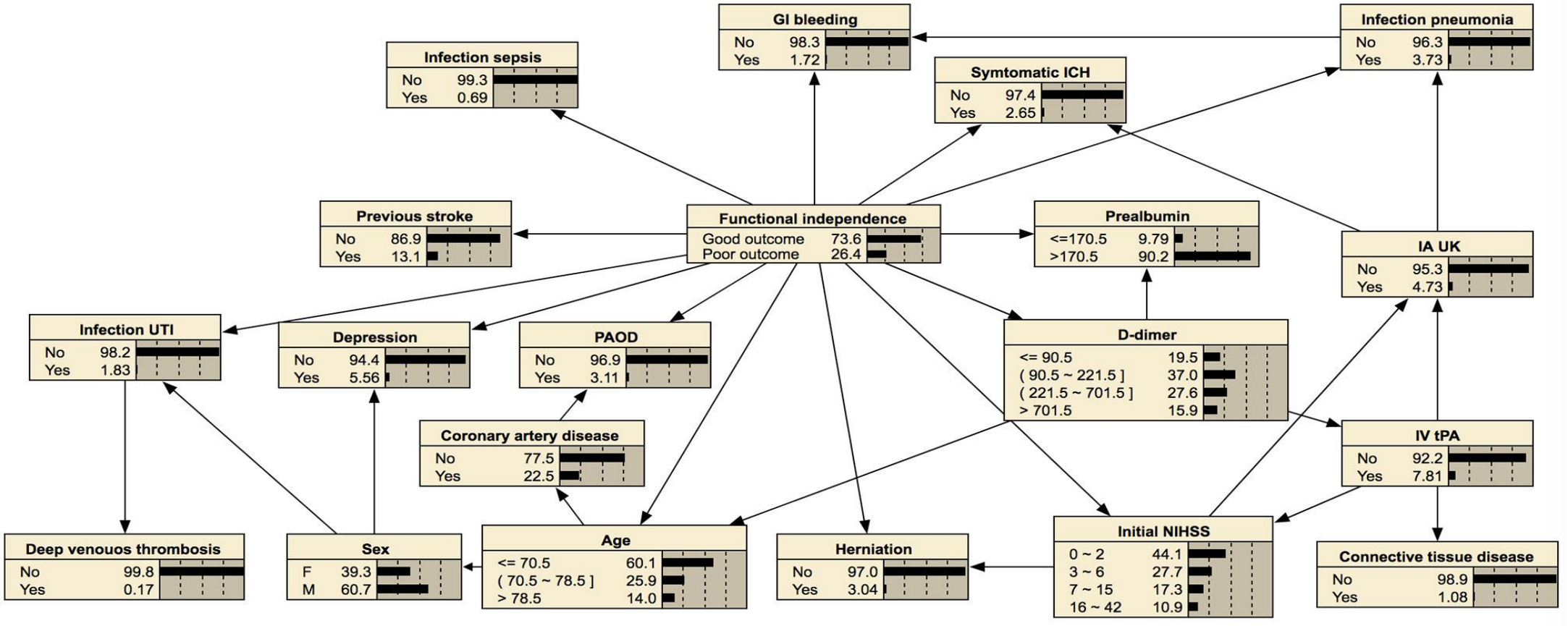
and |B| is the number of parameters in B.

The smaller the MDL score, the better the network.

- For the type of Bayesian network structure, tree-augmented network (TAN) structures were constructed that restrict the number of parents to two nodes

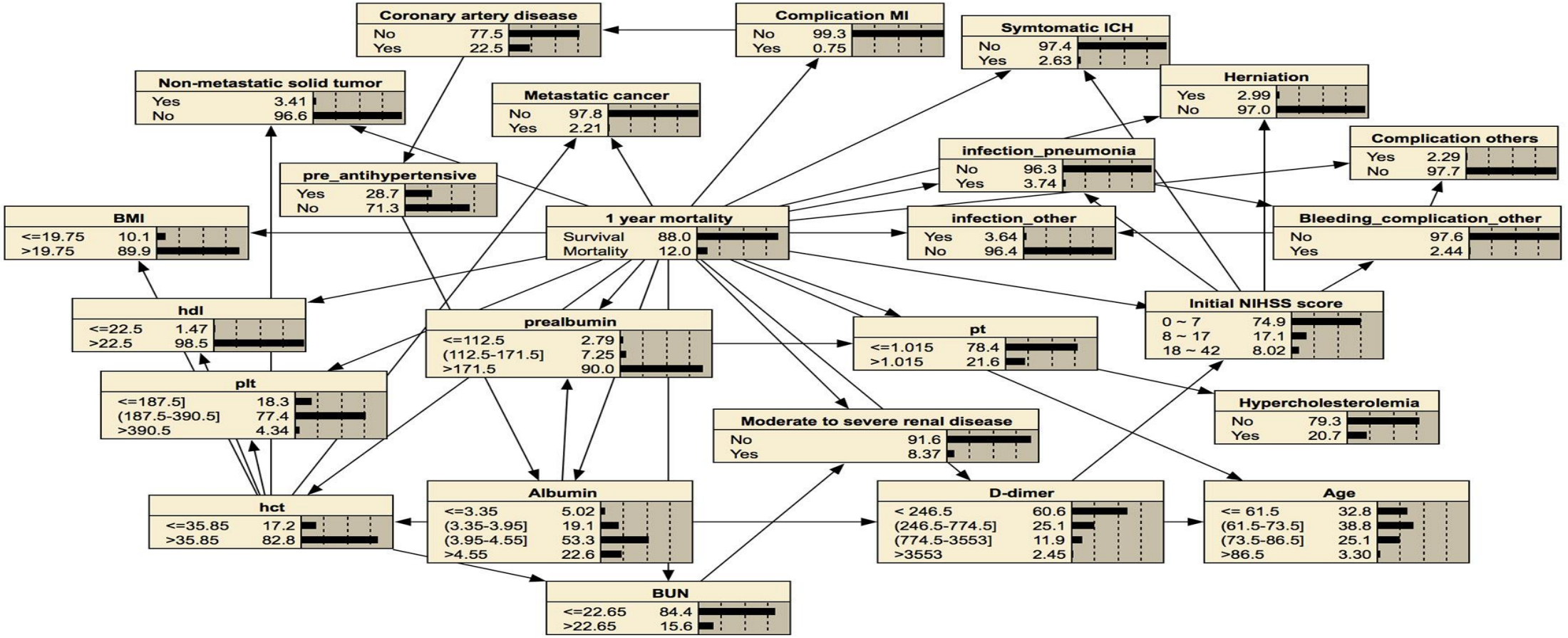

raw data
Yonsei Stroke Registry
4105 records of acute stroke patients
data preparation
Data cleaning
Feature extraction
n=3605, 76 variables predicting death within 1 year
n=3605, 76 variables predicting functional independence at 3-months
feature selection
Information gain
Calculation of entropy
Selection by ranking
Bayesian network
Training a Bayesian network (FS-BN)
Feature selection in cross validation of FS-BN
Selection by subset evaluation
50 variables for 1-year mortality prediction
50 variables for 3-month functional independence
24 variables for 1-year mortality prediction
19 variables for 3-month functional independence
Construction of Bayesian Network Prediction Model & Evaluation
Tree-augmented Bayesian network Structurer learning
Creating candidate Bayesian networks
Calculation of maximum description length
Tree-augmented Bayesian network Parameter learning
Hill-Climbing search
10-cross validation
Bayesian Interchange Format
Construction of Interactive inference system predicting post-stroke outcomes

Results



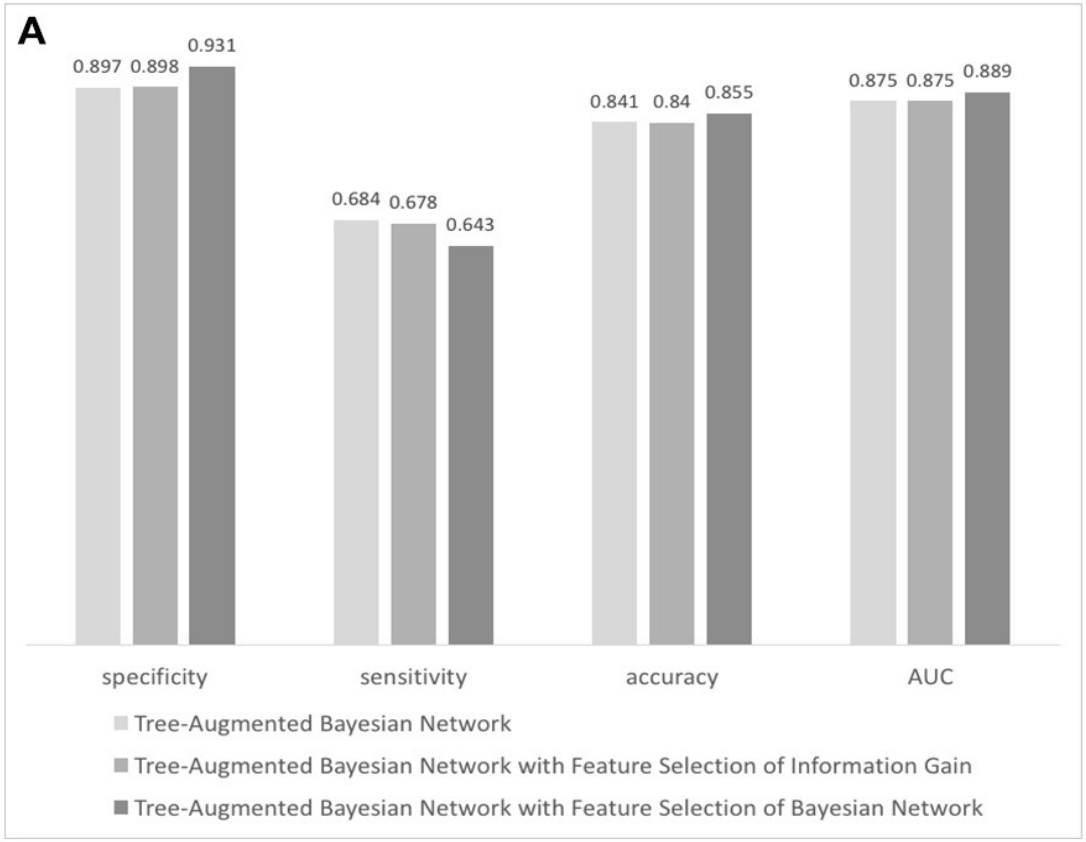
Bayesian network for predicting functional independence at 3 months. The tree-augmented Bayesian network used 19 variables selected by the wrapper of the Bayesian network for prediction.

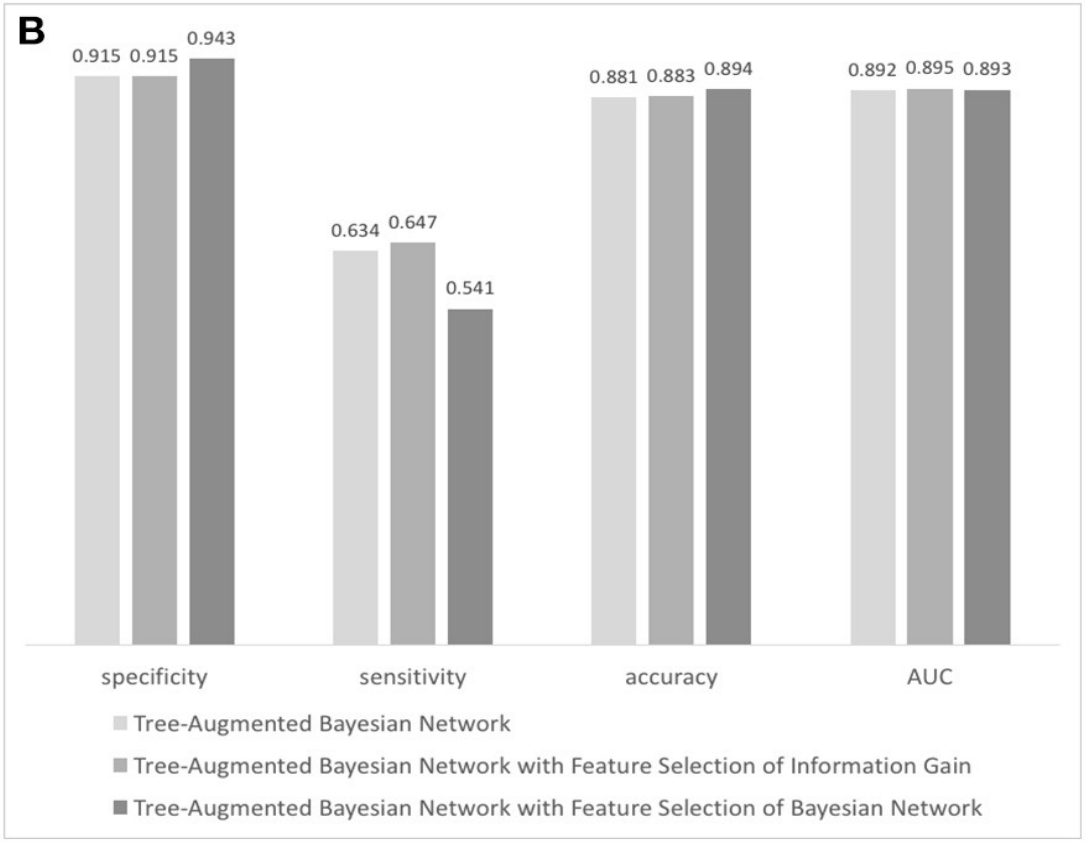
Results Cont.



Bayesian network for predicting 1-year mortality. The tree-augmented Bayesian network used 24 variables selected by the wrapper of the Bayesian network for prediction.

Performance Evaluations





Performance evaluation of Bayesian network-based classifiers: (A) performance of classifiers forecasting 90-day functional independence and (B) performance of classifiers for 1-year mortality prediction.

Programmed Example

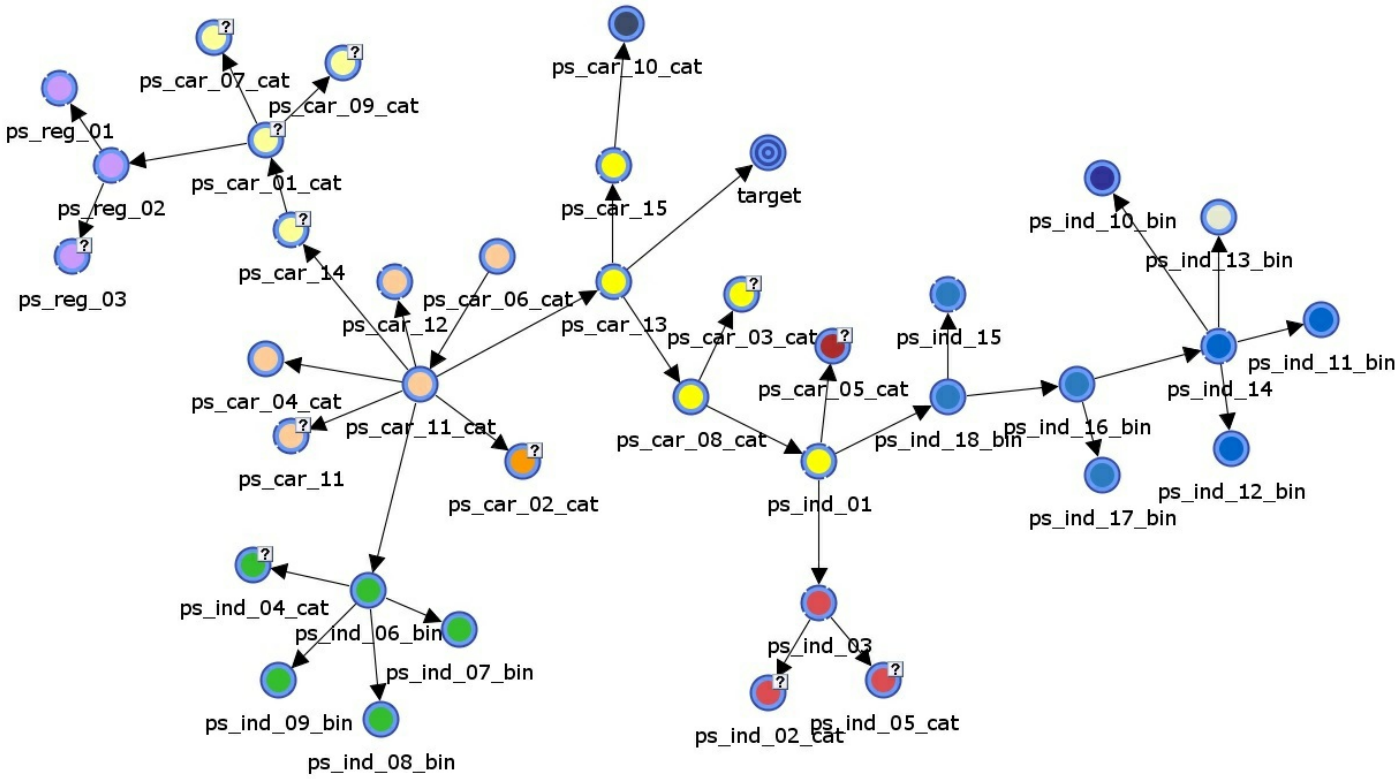
References

- https://www.kaggle.com/johnoliverjones/naive-bayesian-network-with-7-features/comments
- https://www.kaggle.com/c/porto-seguro-safe-driver-prediction
- https://www.cs.ubc.ca/~murphyk/Teaching/CS540-Fall08/L15DGM.pdf
- https://www.frontiersin.org/articles/10.3389/fneur.2018.00699/full

Thank You

Questions/ Queries? Do reach out to me!

We try to find 'relatively' independent features is to first group the features into Concepts. My exploration has shown about 14 Concepts in the data (ps: I excluded all calc features as white noise). To identify Concepts I first learn a BN model of the data with great restrictions on the learning process: features/nodes may have a single parent. Here is a DAG of such a model:



The 14 Concepts are color coded. The small "?" symbol means the feature has missing data. The color coding is imperfect and is is difficult to see that ps_ind_16_bin and ps_ind_14 are different Concepts.

The mutual information of each feature with the target is calculated and the features with at least 0.001 bits of mutual information are selected from each Concept. A total of 11 features were selected and shown below. Sons and Spouses learning resulted in the final 7 feature model.

Importing the libraries

In [ ]:

```
import numpy as np
import pandas as pd
```

Importing the Dataset

In [ ]:

```
train = pd.read_csv('/content/train.csv', usecols = ['target', 'ps_car_07_cat',
    'ps_car_02_cat', 'ps_car_13','ps_reg_02', 'ps_ind_06_bin', 'ps_ind_16_bin', 'ps_ind_
17_bin'])

test = pd.read_csv('/content/test.csv', usecols = ['id', 'ps_car_07_cat', 'ps_car_02_cat
',
    'ps_car_13','ps_reg_02', 'ps_ind_06_bin', 'ps_ind_16_bin', 'ps_ind_17_bin'])
```

In [ ]:

```
train.head(4)
```

Out[ ]:

	target	ps_ind_06_bin	ps_ind_16_bin	ps_ind_17_bin	ps_reg_02	ps_car_02_cat	ps_car_07_cat	ps_car_13
0	0	0	0	1	0.2	1	1	0.883679
1	0	0	0	0	0.4	1	1	0.618817
2	0	0	1	0	0.0	1	1	0.641586
3	0	1	1	0	0.2	1	1	0.542949

In [ ]:

```
test.head(4)
```

Out[ ]:

	id	ps_ind_06_bin	ps_ind_16_bin	ps_ind_17_bin	ps_reg_02	ps_car_02_cat	ps_car_07_cat	ps_car_13
0	0	0	1	0	0.3	1	1	0.669556
1	1	0	1	0	0.5	1	1	0.606320
2	2	0	0	0	0.0	1	1	0.896239
3	3	1	1	0	0.2	1	1	0.652110

Binning the Data

In [ ]:

```
bins = [0.0, 0.639, 0.784, 1.093, 4.4]
train['ps_car_13_d'] = pd.cut(train['ps_car_13'], bins)
test['ps_car_13_d'] = pd.cut(test['ps_car_13'], bins)
bins2 = [-0.1, 0.25, 0.75, 2.0]
train['ps_reg_02_d'] = pd.cut(train['ps_reg_02'], bins2)
test['ps_reg_02_d'] = pd.cut(test['ps_reg_02'], bins2)
train.head(4)
```

Out[ ]:

	target	ps_ind_06_bin	ps_ind_16_bin	ps_ind_17_bin	ps_reg_02	ps_car_02_cat	ps_car_07_cat	ps_car_13	ps_car_13_d	ps_r
0	0	0	0	1	0.2	1	1	0.883679	(0.784, 1.093]	(-(
1	0	0	0	0	0.4	1	1	0.618817	(0.0, 0.639]	(0.
2	0	0	1	0	0.0	1	1	0.641586	(0.639, 0.784]	(-(
3	0	1	1	0	0.2	1	1	0.542949	(0.0, 0.639]	(-(

In [ ]:

```
test.head(4)
```

Out[ ]:

	id	ps_ind_06_bin	ps_ind_16_bin	ps_ind_17_bin	ps_reg_02	ps_car_02_cat	ps_car_07_cat	ps_car_13	ps_car_13_d	ps_reg_(
0	0	0	1	0	0.3	1	1	0.669556	(0.639, 0.784]	(0.25, (
1	1	0	1	0	0.5	1	1	0.606320	(0.0, 0.639]	(0.25, (
2	2	0	0	0	0.0	1	1	0.896239	(0.784, 1.093]	(-0.1, (
3	3	1	1	0	0.2	1	1	0.652110	(0.639, 0.784]	(-0.1, (

Calculated Probabilities

A Naive model was chosen in part because it is so easy to calculate in Python (any language really). It is a simple calculation of the conditional probability of the target given features 1 through 7:

p(target | feature1, ... , feature7) = p(target=1) x { p(feature1 | target) x ... x p(feature7 | target) } / Z

Z is a normalizing constant. It is calculated and used to normalize the whole prediction so that it has an average of about 3.75% (the average probability that target = 1).

In [ ]:

```
# Now calculate the each factor associated with each feature, (fi): p( feature_i | target
)

f1 = pd.DataFrame()
f2 = pd.DataFrame()
f3 = pd.DataFrame()
f4 = pd.DataFrame()
f5 = pd.DataFrame()
f6 = pd.DataFrame()
f7 = pd.DataFrame()

f1 = train.groupby('ps_car_13_d')['target'].agg([('p_f1','mean')]).reset_index()
f2 = train.groupby('ps_reg_02_d')['target'].agg([('p_f2','mean')]).reset_index()
f3 = train.groupby(['ps_car_07_cat'])['target'].agg([('p_f3','mean')]).reset_index()
f4 = train.groupby(['ps_car_02_cat'])['target'].agg([('p_f4','mean')]).reset_index()
f5 = train.groupby('ps_ind_06_bin')['target'].agg([('p_f5','mean')]).reset_index()
f6 = train.groupby('ps_ind_16_bin')['target'].agg([('p_f6','mean')]).reset_index()
f7 = train.groupby('ps_ind_17_bin')['target'].agg([('p_f7','mean')]).reset_index()
```

In [ ]:

```
f1.head(4)
```

Out[ ]:

	ps_car_13_d	p_f1
0	(0.0, 0.639]	0.025037
1	(0.639, 0.784]	0.031396
2	(0.784, 1.093]	0.041144
3	(1.093, 4.4]	0.059181

In [ ]:

```
f2.head(4)
```

Out[ ]:

	ps_reg_02_d	p_f2
0	(-0.1, 0.25]	0.030174
1	(0.25, 0.75]	0.037257
2	(0.75, 2.0]	0.047765

In [ ]:

```
f3.head(4)
```

Out[ ]:

	ps_car_07_cat	p_f3
0	-1	0.078162

	ps_car_07_cat	p_f3
1	0	0.051188
2	1	0.034766

In [ ]:

```
f4.head(4)
```

Out[ ]:

	ps_car_02_cat	p_f4
0	-1	0.000000
1	0	0.049507
2	1	0.033772

In [ ]:

```
f5.head(4)
```

Out[ ]:

	ps_ind_06_bin	p_f5
0	0	0.041585
1	1	0.028537

In [ ]:

```
f6.head(4)
```

Out[ ]:

	ps_ind_16_bin	p_f6
0	0	0.043714
1	1	0.032718

In [ ]:

```
f7.head(4)
```

Out[ ]:

	ps_ind_17_bin	p_f7
0	0	0.033870
1	1	0.055155

Merge all the probabilities with data on 1 big dataframe

In [ ]:

```
sol1 = pd.DataFrame()
sol1 = test.merge(f1, on = 'ps_car_13_d')
sol2 = pd.DataFrame()
sol2 = sol1.merge(f2, on = 'ps_reg_02_d')
del sol1
sol3 = pd.DataFrame()
sol3 = sol2.merge(f3, on = 'ps_car_07_cat')
del sol2
sol4 = pd.DataFrame()
sol4 = sol3.merge(f4, on = 'ps_car_02_cat')
del sol3
```

```
sol5 = pd.DataFrame()
sol5 = sol4.merge(f5, on = 'ps_ind_06_bin')
del sol4
sol6 = pd.DataFrame()
sol6 = sol5.merge(f6, on = 'ps_ind_16_bin')
del sol5
sol = pd.DataFrame()
sol = sol6.merge(f7, on = 'ps_ind_17_bin')
del sol6
sol.head(5)
```

Out[ ]:

	id	ps_ind_06_bin	ps_ind_16_bin	ps_ind_17_bin	ps_reg_02	ps_car_02_cat	ps_car_07_cat	ps_car_13	ps_car_13_d	ps_reg
0	0	0	1	0	0.3	1	1	0.669556	(0.639, 0.784]	(0.25
1	94	0	1	0	0.6	1	1	0.756009	(0.639, 0.784]	(0.25
2	167	0	1	0	0.7	1	1	0.729712	(0.639, 0.784]	(0.25
3	240	0	1	0	0.6	1	1	0.756472	(0.639, 0.784]	(0.25
4	276	0	1	0	0.5	1	1	0.720408	(0.639, 0.784]	(0.25

Computation

In [ ]:

```
# f is the product of factors of feaures
sol.loc[:,'f'] = sol.loc[:,'p_f1'] * sol.loc[:,'p_f2'] * sol.loc[:,'p_f3'] * sol.loc[:,'p
_f4'] \
                * sol.loc[:,'p_f5'] * sol.loc[:,'p_f6'] * sol.loc[:,'p_f7']

z = sol.f.sum() / len(sol.f)
# z is the normalizing factor
sol['target'] = 0.03645 * sol.loc[:,'f'] / z
#sol[['id', 'target']].to_csv('bn_5_output_7_nodes.csv', index = False, float_format='%.4
f')
```

In [ ]:

```
sol[['id', 'target']]
```

Out[ ]:

	id	target
0	0	0.024344
1	94	0.024344
2	167	0.024344
3	240	0.024344
4	276	0.024344
...	...	...
892811	1303796	0.289477
892812	1382185	0.289477
892813	1390497	0.289477
892814	1400101	0.289477
892815	1486299	0.289477

892816 rows × 2 columns

Performance Metrics (Compute GINI)

In [ ]:

```
from numba import jit
```

In [ ]:

```
@jit
def eval_gini(y_true, y_prob):
    y_true = np.asarray(y_true)
    y_true = y_true[np.argsort(y_prob)]
    ntrue = 0
    gini = 0
    delta = 0
    n = len(y_true)
    for i in range(n-1, -1, -1):
        y_i = y_true[i]
        ntrue += y_i
        gini += y_i * delta
        delta += 1 - y_i
    gini = 1 - 2 * gini / (ntrue * (n - ntrue))
    return gini
```

In [ ]:

```
sol1 = pd.DataFrame()
sol1 = train.merge(f1, on = 'ps_car_13_d')
sol2 = pd.DataFrame()
sol2 = sol1.merge(f2, on = 'ps_reg_02_d')
del sol1
sol3 = pd.DataFrame()
sol3 = sol2.merge(f3, on = 'ps_car_07_cat')
del sol2
sol4 = pd.DataFrame()
sol4 = sol3.merge(f4, on = 'ps_car_02_cat')
del sol3
sol5 = pd.DataFrame()
sol5 = sol4.merge(f5, on = 'ps_ind_06_bin')
del sol4
sol6 = pd.DataFrame()
sol6 = sol5.merge(f6, on = 'ps_ind_16_bin')
del sol5
sol = pd.DataFrame()
sol = sol6.merge(f7, on = 'ps_ind_17_bin')
del sol6
sol.loc[:,'f'] = sol.loc[:,'p_f1'] * sol.loc[:,'p_f2'] * sol.loc[:,'p_f3'] * sol.loc[:,'p
_f4'] \
                * sol.loc[:,'p_f5'] * sol.loc[:,'p_f6'] * sol.loc[:,'p_f7']
z = sol.f.sum() / len(sol.f)
sol['exp_target'] = 0.03645 * sol.loc[:,'f'] / z

# Calculate GINI score
eval_gini(sol['target'], sol['exp_target'])
```

```
<ipython-input-11-e1bbcc8b7298>:1: NumbaWarning:
Compilation is falling back to object mode WITH looplifting enabled because Function "eva
l_gini" failed type inference due to: non-precise type pyobject
During: typing of argument at <ipython-input-11-e1bbcc8b7298> (3)

File "<ipython-input-11-e1bbcc8b7298>", line 3:
def eval_gini(y_true, y_prob):
    y_true = np.asarray(y_true)
    ^

  @jit
<ipython-input-11-e1bbcc8b7298>:1: NumbaWarning:
```

```
Compilation is falling back to object mode WITHOUT looplifting enabled because Function "
eval_gini" failed type inference due to: cannot determine Numba type of <class 'numba.cor
e.dispatcher.LiftedLoop'>

File "<ipython-input-11-e1bbcc8b7298>", line 9:
def eval_gini(y_true, y_prob):
    <source elided>
    n = len(y_true)
    for i in range(n-1, -1, -1):
    ^

  @jit
/usr/local/lib/python3.7/dist-packages/numba/core/object_mode_passes.py:178: NumbaWarning
: Function "eval_gini" was compiled in object mode without forceobj=True, but has lifted
loops.

File "<ipython-input-11-e1bbcc8b7298>", line 3:
def eval_gini(y_true, y_prob):
    y_true = np.asarray(y_true)
    ^

  state.func_ir.loc))
/usr/local/lib/python3.7/dist-packages/numba/core/object_mode_passes.py:188: NumbaDepreca
tionWarning:
Fall-back from the nopython compilation path to the object mode compilation path has been
detected, this is deprecated behaviour.

For more information visit https://numba.pydata.org/numba-doc/latest/reference/deprecatio
n.html#deprecation-of-object-mode-fall-back-behaviour-when-using-jit

File "<ipython-input-11-e1bbcc8b7298>", line 3:
def eval_gini(y_true, y_prob):
    y_true = np.asarray(y_true)
    ^

  state.func_ir.loc))
```

Out[ ]:

```
0.21083658131836647
```

How to do this in Spark?

Create a Spark Dataframe and then pass it as a UDF, where all functionality of Pandas can be used inside the UDF.

frontiers in Neurology

ORIGINAL RESEARCH
published: 07 September 2018
doi: 10.3389/fneur.2018.00699

A Bayesian Network Model for Predicting Post-stroke Outcomes With Available Risk Factors

Eunjeong Park[1], *Hyuk-jae Chang*[2] *and Hyo Suk Nam*[3]*

[1] Cardiovascular Research Institute, College of Medicine, Yonsei University, Seoul, South Korea, [2] Department of Cardiology, College of Medicine, Yonsei University, Seoul, South Korea, [3] Department of Neurology, College of Medicine, Yonsei University, Seoul, South Korea

Bayesian network is an increasingly popular method in modeling uncertain and complex problems, because its interpretability is often more useful than plain prediction. To satisfy the core requirement in medical research to obtain interpretable prediction with high accuracy, we constructed an inference engine for post-stroke outcomes based on Bayesian network classifiers. The prediction system that was trained on data of 3,605 patients with acute stroke forecasts the functional independence at 3 months and the mortality 1 year after stroke. Feature selection methods were applied to eliminate less relevant and redundant features from 76 risk variables. The Bayesian network classifiers were trained with a hill-climbing searching for the qualified network structure and parameters measured by maximum description length. We evaluated and optimized the proposed system to increase the area under the receiver operating characteristic curve (AUC) while ensuring acceptable sensitivity for the class-imbalanced data. The performance evaluation demonstrated that the Bayesian network with selected features by wrapper-type feature selection can predict 3-month functional independence with an AUC of 0.889 using only 19 risk variables and 1-year mortality with an AUC of 0.893 using 24 variables. The Bayesian network with 50 features filtered by information gain can predict 3-month functional independence with an AUC of 0.875 and 1-year mortality with an AUC of 0.895. We also built an online prediction service, Yonsei Stroke Outcome Inference System, to substantialize the proposed solution for patients with stroke.

Keywords: stroke, bayesian network, prognostic model, machine learning classification, decision support techniques, imbalanced data

OPEN ACCESS

Edited by:
Fabien Scalzo, University of California, Los Angeles, United States

Reviewed by:
Jens Fiehler, Universitätsklinikum Hamburg-Eppendorf, Germany
Katharina Stibrant Sunnerhagen, University of Gothenburg, Sweden

***Correspondence:**
Hyo Suk Nam
hsnam@yuhs.ac

Specialty section:
This article was submitted to Stroke, a section of the journal Frontiers in Neurology

Received: *30 May 2018*
Accepted: *02 August 2018*
Published: *07 September 2018*

Citation:
Park E, Chang H-j and Nam HS (2018) A Bayesian Network Model for Predicting Post-stroke Outcomes With Available Risk Factors. Front. Neurol. 9:699. doi: 10.3389/fneur.2018.00699

INTRODUCTION

A stroke is the second most common cause of death in the world and a leading cause of long-term disability. Patients with stroke have higher mortality than age- and sex-matched subjects who have not experienced a stroke. It is also reported that strokes recur in 6–20% of patients, and approximately two-thirds of stroke survivors continue to have functional deficits that are associated with diminished quality of life (1). Such disability after stroke can be measured by the modified Rankin scale that categorizes functional ability from 0 to 6 (2–4). To discriminate the effect of clinical treatment for patients with ischemic stroke, a score on the modified Rankin scale 0–2 is widely applied for the indication of functional independence after stroke (2).

There are many prognostic models for the functional outcomes and risk of death after stroke. However, an agreed set of guidelines or reporting for the development of prognostic score models are currently unavailable. In a recent systematic review of clinical prediction models, the discriminative performances of models were still unsatisfactory, with the AUC values ranging from 0.60 to 0.72, which are similar to the predictability of experienced clinicians (5).

The prediction of prognosis needs to employ a variety of statistical, probabilistic, and optimization techniques to learn patterns from large, complex, and unbalanced medical data. This complexity challenges researchers to apply machine learning techniques to diagnose and predict the progress of the disease (6, 7). Machine learning has been expected to dramatically improve prognosis, and certain applications have achieved remarkable results (7). These applications have employed various machine learning techniques including a deep neural network (8), support vector machine (8, 9), decision trees (10), and ensemble methods (11, 12) to classify diseases, level of deficits, and morality. Selecting the optimal solution for a decision problem should consider the unique pattern of a data set and the specific characteristics of the problem (13).

The Bayesian network, a machine learning method, predicts and describes classification based on the Bayes theorem (14). Bayesian networks are widely used in medical decision support for their ability to intuitively encapsulate cause and effect relationships between factors that are stored in medical data (15, 16). With these characteristics of conditional probabilities, the Bayesian network can provide interpretable classifiers by logic inherent in a decision support (17, 18). The parameters and their dependences with conditional probabilities of the Bayesian network can be provided either by experts' knowledge (16, 19) or by automatic learning from data (20, 21). In addition, Bayesian networks can be used to query any given node in the network and are therefore substantially more useful in clinics compared with classifiers built based on specific outcome variables (22).

In this study, our aim was to investigate the usefulness of a machine learning method to forecast functional recovery for independent activities and 1-year mortality in patients with acute ischemic stroke. We also introduced an online inference system for predicting functional independence at 3 months and mortality in 1 year of patients with stroke based on the proposed Bayesian network.

MATERIALS AND METHODS

Data Set

Subjects for this study were selected from consecutive patients with acute ischemic stroke who had been registered in the Yonsei Stroke Registry over a 6.5-year period (January 2007 to June 2013). The Yonsei Stroke Registry is a prospective hospital-based registry for patients with acute ischemic stroke or transient ischemic attack within 7 days after symptom onset (23).

During admission, all patients were thoroughly investigated for medical history, clinical manifestations, and the presence of vascular risk factors. Every patient was evaluated with 12-lead electrocardiography, chest x-ray, lipid profiles, and standard blood tests. All registered patients underwent brain imaging studies including brain computed tomography (CT) and/or MRI. Angiographic studies using CT angiography, magnetic resonance angiography, or digital subtraction angiography were included in the standard evaluation. Additional blood tests for coagulopathy or prothrombotic conditions were performed in patients younger than 45 years. Transesophageal echocardiography was included in the standard evaluation, except in patients with decreased consciousness, impending brain herniation, poor systemic condition, inability to accept an esophageal transducer because of swallowing difficulty or tracheal intubation, or lack of informed consent (24). Transthoracic echocardiography, heart CT, and Holter monitoring were also performed in selected patients (25). When a patient was admitted more than twice because of recurrent strokes, only data for the first admission were used for this study. Initial stroke severity was determined by National Institute of Health Stroke Scale (NIHSS) scores and score tertiles were used for the analysis.

Hypertension was defined as resting systolic blood pressure $\geq$140 mm Hg or diastolic blood pressure $\geq$90 mm Hg after repeated measurements during hospitalization or currently taking antihypertensive medication. Diabetes mellitus was defined as fasting plasma glucose values $\geq$7 mmol/L or taking an oral hypoglycemic agent or insulin. Hyperlipidemia was diagnosed as a fasting serum total cholesterol level $\geq$6.2 mmol/L, low-density lipoprotein cholesterol $\geq$4.1 mmol/L, or currently taking a lipid-lowering drug after a hyperlipidemia diagnosis. A current smoker was defined as an individual who smoked at the time of stroke or had quit smoking 1 year before treatment (26). The collection of variables during admission including clinical, imaging, and laboratory data were used in statistical analysis and Bayesian network modeling.

Stroke classification was determined during weekly conferences based on the consensus of stroke neurologists. Data including clinical information, risk factors, imaging study findings, laboratory analyses, and other special evaluations were collected. Along with these data, prognosis during hospitalization and long-term outcomes were also determined. Data were entered into a web-based registry. Stroke subtypes were identified according to the Trial of ORG 10172 in Acute Stroke Treatment (TOAST) classification (27).

For target variables in classification, we collected the outcome variables for patients who were followed in the outpatient clinic or by a structured telephone interview at 3 months and every year after discharge. Short-term functional outcomes at 3 months were determined based on the modified Rankin scale. Major disability was defined as a score on the modified Rankin scale of 3–6, as a poor outcome at 3 months after stroke. Deaths among subjects from January 2001 to December 31, 2013, were confirmed by matching the information in the death records and identification numbers assigned to the subjects at birth (5). We obtained data for the date and causes of death from the Korean National Statistical Office, which were identified based on death certificates (28, 29). The institutional review board of Severance Hospital, Yonsei University Health System, approved this study and waived the patients' informed consent

because of a retrospective design and observational nature of this study.

Bayesian Networks

The collected data set was used to construct Bayesian networks for predicting post-stroke outcomes. We extracted a total of 76 random variables of each instance for patient data. A Bayesian network consists of a directed acyclic graph whose nodes represent random variables and links express dependences between nodes. Suppose random variables $V_i \in V\,(1 \leq i \leq n)$. A Bayesian network is described as a directed acyclic graph $G = (V, A, P)$ with links $A \subseteq V \times V$ and P a joint probability distribution. P, a joint probability over V, is described as

$$P(V) = \prod_{V_i \in V} P(V_i \,|\pi(V_i)),$$

where $\pi(V_i)$ is the set of parent nodes of V_i.

Training Bayesian network classifiers is the process of parameter learning to find optimal Bayesian structures estimating parameter set of P that best represents given data set with labeled instances (13). Given a data set D with variable V_i, the observed distribution P_D is described as a joint probability distribution over D. The learning process now measures and compares the quality of Bayesian networks to evaluate how well the represented distribution explains the given data set. The log-likelihood is the basic common value used for measuring the quality of a Bayesian network as follows:

$$LL(\mathcal{B}\,|D) = \sum\nolimits_{V_i} log(P(V_i \,|\pi_{\mathcal{B}}(V_i))),$$

where $\mathcal{B}$ is the Bayesian network over D and $|\pi_B(V_i)$ is parent nodes of V_i in $\mathcal{B}$(13, 30).

Diverse quality measurement methods have been investigated (31). The algorithm searched the best Bayesian network based on the Bayesian information criterion (32), Bayesian Dirichlet equivalence score (19), Akaike information criterion (AIC) (33), and the maximum description length (MDL) scores (30, 34). In this study, we used the MDL score to evaluate the quality of a Bayesian network. The MDL score is described as

$$MDL = -LL(\mathcal{B}\,|D) + \frac{log\,N}{2} \cdot |\mathcal{B}|,$$

where N is the number of instances in D, and $|\mathcal{B}|$ is the number of parameters in $\mathcal{B}$. The smaller the MDL score, the better the network. The search algorithm, greedy hill-climbing algorithm (35) in our study, selects the best Bayesian network by calculating MDL scores of candidate networks. For the type of Bayesian network structure, we constructed tree-augmented network (TAN) structures that restrict the number of parents to two nodes (36).

Prediction Process

The entire process of a Bayesian network-based prediction system is shown in **Figure 1**. A total of 76 features were extracted from the Yonsei Stroke Registry and data preparation process filtered records with missing outcome variables and exclusion criteria. For feasible prediction service in clinical environment, we performed two different feature selection methods.

Feature selection or dimension reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables (37, 38). Feature selection improves the overfitting problem caused by irrelevant or redundant variables that may strongly bias the performance of the classifier. The definition of feature selection in formal expression is described in Drugan and Wiering (30) and Hruschka et al. (39). In many studies, feature selection methods are categorized into filters, wrappers, or embedded methods that are applied to the data set in advance of the training learning algorithm, or to embed feature selection in the learning process (37, 40). Filter methods select features based on a performance measure regardless of the employed data modeling algorithm. The filter approach selects random variables based on information gain score, ReliefF, or correlation-based method by ranking variables or searching subset of variables. Information gain measures the amount of entropy as a measure of uncertainty reduced by knowing a feature (41–43); ReliefF evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and the different class (44, 45); and correlation evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them (46, 47). Unlike the filter approach, wrapper methods measure the usefulness of a subset of features by actually training a model on it. We evaluated the performance of Bayesian networks with a reduced variable set selected by information gain and Bayesian network algorithms that are popular in filter and wrapper methods (42, 48, 49).

First, we tested the Bayesian network classifier with features chosen by information gain based on entropy of each feature. The other feature selection method, considering the characteristics of Bayesian network classifiers, reduces the variable set by evaluating the performance of the Bayesian network classifier in cross-validation in which a search algorithm extracts a subset of attributes to maximize AUC in prediction (**Figure 1**). The optimization for AUC is to solve the imbalance between the number of survival and mortal subjects.

Using the reduced variables by feature selection, the system constructed a Bayesian network prediction model to search optimal Bayesian network structures and parameters. We evaluated the performance of prediction algorithms using (1) a basic tree-augmented Bayesian network, (2) a tree-augmented Bayesian network with features filtered by information gain, and (3) a tree-augmented Bayesian network with features filtered by the wrapper of a Bayesian network. The performances of all Bayesian networks and predictive models were evaluated based on the AUC, specificity, and sensitivity of 10-fold cross-validations (50). We also implemented an online prediction system for post-stroke outcomes embedding the trained classifiers. In the validation process, we bound the minimum sensitivity as 0.50 to utilize the trained classifiers in real-world applications with imbalanced data.

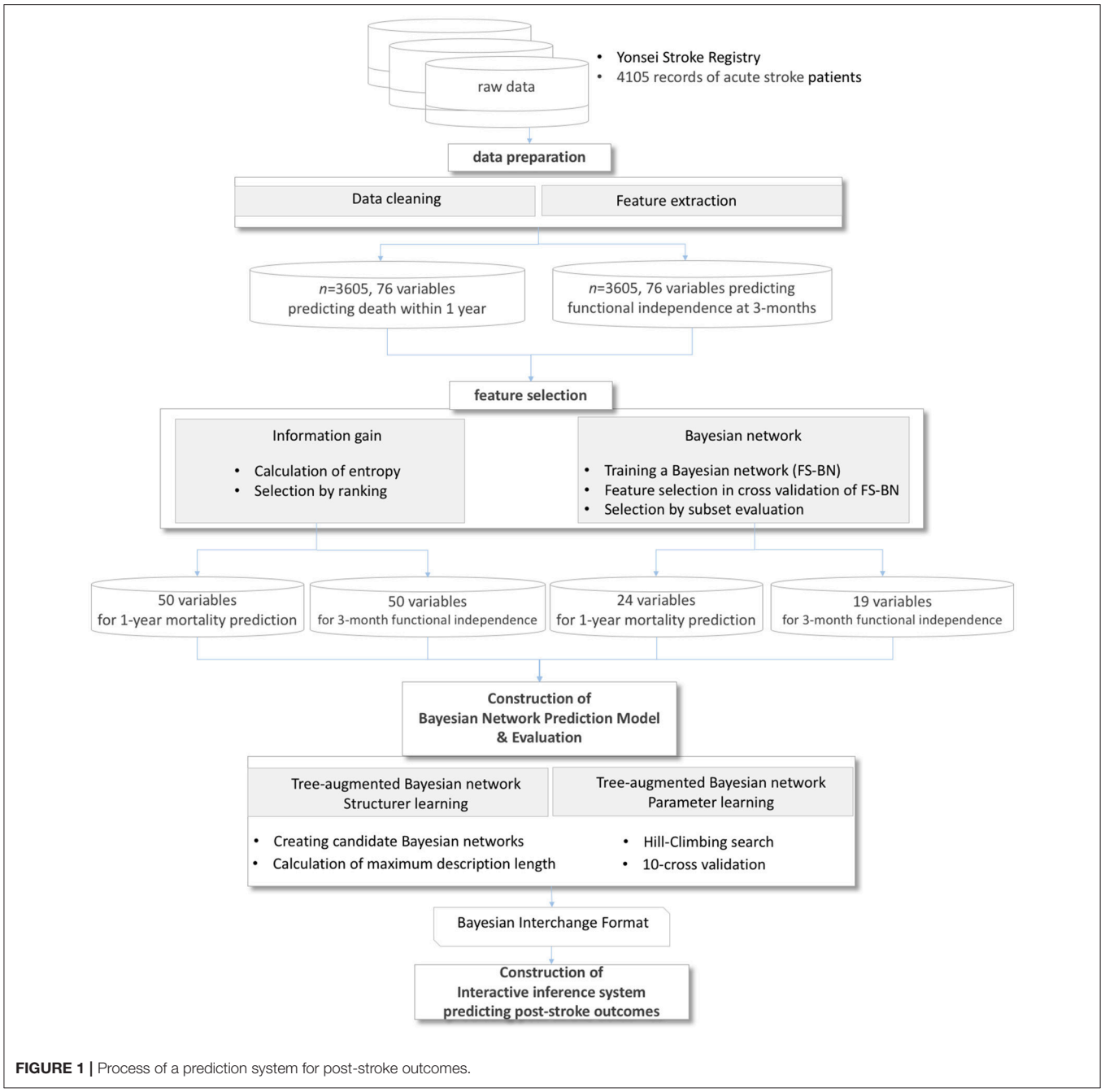


FIGURE 1 | Process of a prediction system for post-stroke outcomes.

RESULTS

Statistical Characteristics

During the study period, 4,105 consecutive patients with acute ischemic stroke or transient ischemic attack were registered to the Yonsei Stroke Registry. Exclusion criteria of this study were patients with the stroke subtypes other than cryptogenic stroke including transient ischemic attack ($n = 326$), foreigner ($n = 48$), missing data ($n = 29$), follow-up loss ($n = 97$). After exclusion, a total of 3,605 patients were finally enrolled for this study. The mean age was 65.9 ± 12.6 years, and 60.7% were men. A comparison of demographic characteristics between the outcome at 3 months and death within 1 year is shown at **Table 1**. Patients with poor outcome were older, more likely to be women, not a current smoker, frequently had old stroke, hypertension, atrial fibrillation, congestive heart failure, peripheral artery obstructive disease, or anemia. Thrombolysis or endovascular mechanical thrombectomy, symptomatic intracranial hemorrhage, and herniation are frequent in patients with poor outcome. Laboratory data showed that patients with poor outcome showed lower hemoglobin, hematocrit, albumin, prealbumin, body weight and higher ESR, fibrinogen, hsCRP, and D-dimer level. The differences of demographics of patients between survival and

TABLE 1 | Demographic characteristics and comparison of outcome at 3 months and death within 1 year.

	Total	Outcome at 3 months			Death within 1 year		
	(*N* = 3,605)	Good outcome (*N* = 2,653)	Poor outcome (*N* = 952)	*p*	No (*N* = 3,171)	Yes (*N* = 434)	*p*
Age	65.9 ± 12.6	64.0 ± 12.3	71.2 ± 11.9	<0.001	64.8 ± 12.4	73.9 ± 11.2	<0.001
Sex				<0.001			0.016
F	1,416 (39.3%)	969 (36.5%)	447 (47.0%)		1,222 (38.5%)	194 (44.7%)	
M	2,189 (60.7%)	1,684 (63.5%)	505 (53.0%)		1,949 (61.5%)	240 (55.3%)	
Hypertension	2,675 (74.2%)	1,940 (73.1%)	735 (77.2%)	0.015	2,675 (74.2%)	1,940 (73.1%)	0.023
Diabetes	1,144 (31.7%)	827 (31.2%)	317 (33.3%)	0.243	1,144 (31.7%)	827 (31.2%)	0.282
Hypercholesterolemia	747 (20.7%)	554 (20.9%)	193 (20.3%)	0.726	685 (21.6%)	62 (14.3%)	0.001
Current smoking	856 (23.7%)	704 (26.5%)	152 (16.0%)	<0.001	856 (23.7%)	704 (26.5%)	<0.001
Old stroke	472 (13.1%)	301 (11.3%)	171 (18.0%)	<0.001	401 (12.6%)	71 (16.4%)	0.038
Atrial fibrillation	813 (22.6%)	482 (18.2%)	331 (34.8%)	<0.001	623 (19.6%)	190 (43.8%)	<0.001
Coronary artery disease	811 (22.5%)	603 (22.7%)	208 (21.8%)	0.608	717 (22.6%)	94 (21.7%)	0.701
Congestive heart failure	184 (5.1%)	110 (4.1%)	74 (7.8%)	<0.001	134 (4.2%)	50 (11.5%)	<0.001
Peripheral artery obstructive disease	110 (3.1%)	60 (2.3%)	50 (5.3%)	<0.001	85 (2.7%)	25 (5.8%)	0.001
Initial NIHSS score	5.6 ± 6.3	3.4 ± 4.0	11.5 ± 7.5	<0.001	4.8 ± 5.4	11.5 ± 8.4	<0.001
TOAST				<0.001			<0.001
LAC	321 (8.9%)	285 (10.7%)	36 (3.8%)		312 (9.8%)	9 (2.1%)	
LAA	741 (20.6%)	504 (19.0%)	237 (24.9%)		661 (20.8%)	80 (18.4%)	
CE	991 (27.5%)	688 (25.9%)	303 (31.8%)		823 (26.0%)	168 (38.7%)	
SOD	89 (2.5%)	68 (2.6%)	21 (2.2%)		80 (2.5%)	9 (2.1%)	
UT	668 (18.5%)	498 (18.8%)	170 (17.9%)		587 (18.5%)	81 (18.7%)	
UN	785 (21.8%)	607 (22.9%)	178 (18.7%)		703 (22.2%)	82 (18.9%)	
UI	10 (0.3%)	3 (0.1%)	7 (0.7%)		5 (0.2%)	5 (1.2%)	
Anemia	617 (17.1%)	361 (13.6%)	256 (26.9%)	<0.001	450 (14.2%)	167 (38.5%)	<0.001
Thrombolysis	485 (13.5%)	272 (10.3%)	213 (22.4%)	<0.001	377 (11.9%)	108 (24.9%)	<0.001
Symtomatic ICH	92 (2.6%)	10 (0.4%)	82 (8.6%)	<0.001	43 (1.4%)	49 (11.3%)	<0.001
Herniation	105 (2.9%)	3 (0.1%)	102 (10.7%)	<0.001	38 (1.2%)	67 (15.4%)	<0.001
Body weight	62.9 ± 11.1	64.0 ± 10.9	60.0 ± 11.2	<0.001	63.6 ± 11.0	57.8 ± 10.8	<0.001
hgb	13.8 ± 2.0	14.0 ± 1.9	13.3 ± 2.2	<0.001	14.0 ± 1.9	12.7 ± 2.3	<0.001
hct	40.6 ± 5.6	41.1 ± 5.3	39.3 ± 6.1	<0.001	41.0 ± 5.3	37.9 ± 6.5	<0.001
esr	23.9 ± 22.2	21.2 ± 20.1	31.3 ± 25.8	<0.001	22.1 ± 20.6	36.5 ± 28.8	<0.001
pt	1.0 ± 0.5	1.0 ± 0.3	1.0 ± 0.7	0.123	1.0 ± 0.5	1.0 ± 0.2	0.002
Albumin	4.2 ± 0.5	4.3 ± 0.4	4.0 ± 0.5	<0.001	4.3 ± 0.4	3.9 ± 0.6	<0.001
Prealbumin	223.7 ± 72.6	239.0 ± 69.9	205.6 ± 71.6	<0.001	233.3 ± 69.8	186.8 ± 71.4	<0.001
Fibrinogen	322.8 ± 94.3	316.1 ± 83.9	341.5 ± 116.8	<0.001	320.1 ± 88.5	342.5 ± 128.5	0.001
hsCRP	11.3 ± 48.4	7.5 ± 49.7	22.2 ± 42.7	<0.001	9.2 ± 48.4	27.3 ± 45.5	<0.001
D-dimer	779.0 ± 3846.1	418.4 ± 1704.4	1788.2 ± 6834.6	<0.001	464.5 ± 1759.3	3079.8 ± 9723.3	<0.001

death within 1 year were similar with functional outcome at 3 months. D-dimer levels were significantly higher in patients who died within 1 year compared with survivors (3079.8 ± 9723.3 vs. 464.5 ± 1759.3, $p < 0.001$).

Structure and Parameters of Bayesian Networks

As we described in **Figure 1**, two different feature selection techniques were performed in our experiment: variables selected by information gain with ranking or variables selected by a wrapper embedding Bayesian network with greedy stepwise subset selection in cross-validation. The top-ranked variables in the filter by information gain and the wrapper of the Bayesian network in forecasting functional independence at 3 months are shown in **Figures 2A,B**, and variables for predicting 1-year mortality are shown in **Figures 2C,D**. The most affective factor for functional recovery prediction was Initial NIHSS, while D-dimer ranked top in 1-year mortality prediction. The common variables for predicting post-stroke outcomes were Initial NIHSS, D-dimer, hsCPR, and Age. However, the subset-searching algorithm selects a method differently from the

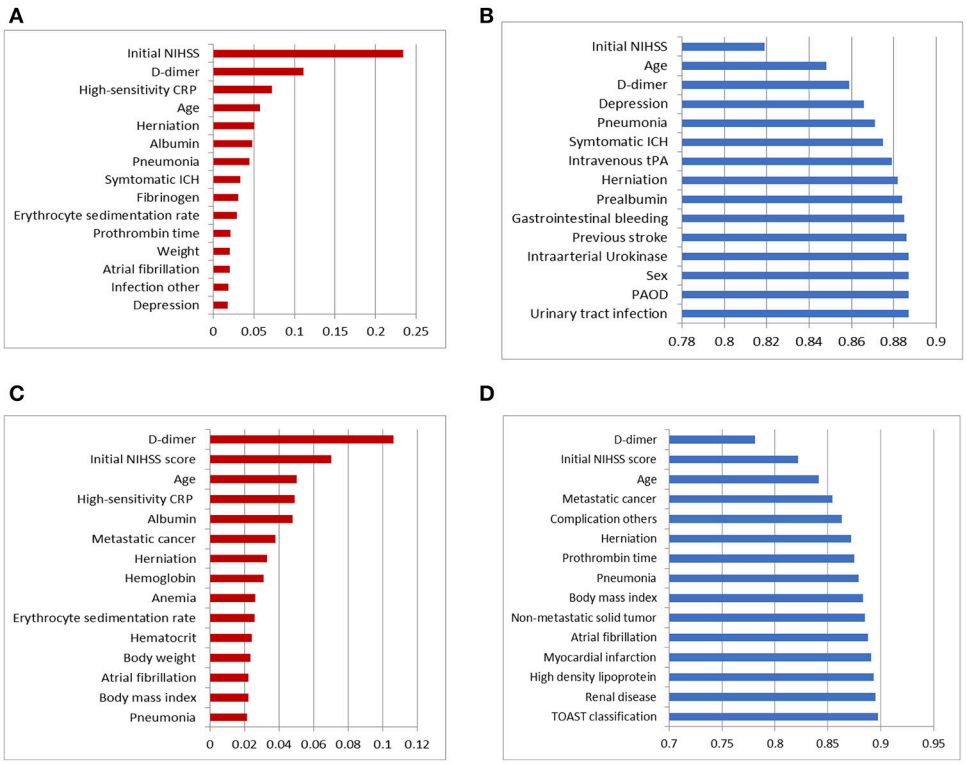


FIGURE 2 | Top 15 variables in dimension reduction for post-stroke outcome prediction: **(A)** variables filtered by ranks of information gain for predicting functional independence at 3 months, **(B)** variables selected by the wrapper of the Bayesian network classifier with greedy subset selection for predicting functional independence at 3 months, **(C)** variables filtered by ranks of information gain for predicting 1-year mortality, and **(D)** variables selected by the wrapper of the Bayesian network classifier with greedy subset selection for predicting 1-year mortality.

ranking method that evaluates the individual variables separately; thus, certain variables were excluded from the selected subset even though their ranks are high in individual evaluation.

Using the result of feature selection, we trained three tree-augmented Bayesian network classifiers; (1) Tree-augmented Bayesian network with the entire dataset, (2) tree-augmented Bayesian network with features filtered by ranking of information gain, and (3) tree-augmented Bayesian network with features filtered by the wrapper of the Bayesian network classifier (see **Figure 3**). The predictive performance for 3-month outcomes is shown in **Figure 3A**. The classifier trained with features chosen by the Bayesian network's subset evaluation performs in prediction of 3-month functional recovery with the specificity of 0.931, accuracy of 0.643, and AUC of 0.889 (95% CI, 0.879–0.899) although the sensitivity (0.643) is slightly lower than other algorithms. The tree-augmented Bayesian network without feature selection achieved the AUC of 0.875 (95% CI, 0.864–0.886), but the highest sensitivity of 0.684; and the Bayesian network with features by ranking of information gain obtained the AUC of 0.875 (95% CI, 0.864–0.886) and mid-level performance between two other algorithms. The Bayesian network classifier with feature selection achieved best performance in most metrics except sensitivity, although it reduced the variable set from 76 variables to 19 variables, resulting in a great reduction in model construction time.

In prediction of 1-year mortality, AUCs of three algorithms were not significantly different (0.892 with 95% CI, 0.872–0.912; 0.895 with CI, 0.875–0.915; and 0.893 with CI, 0.873–0.913). All algorithms achieved higher specificities in predicting 1-year mortality than those for the prediction of functional independence (0.915 vs. 0.897 with a basic Bayesian network, 0.915 vs. 0.898 with a Bayesian network with features filtered by information gain, and 0.943 vs. 0.931 with a Bayesian network with features chosen by the wrapper of the Bayesian network classifier). The Bayesian network algorithm with feature selection for 1-year mortality cuts out the entire variable set to 24 variables that curtail network construction time. The final Bayesian networks predicting functional recovery and 1-year mortality are shown in **Figures 4**, **5**, respectively.

Online Interactive System for Predicting Post-stroke Outcomes

To realize decision support using Bayesian network classifiers, we embedded our final Bayesian networks into an online inference system, Y-SOIS (Yonsei-Stroke Outcome Inference System, https://www.hed.cc/?a=Yonsei_SOIS), that enables answering post-stroke outcomes when users provide available risk variables. **Figure 6** shows the screenshots of Y-SOIS.

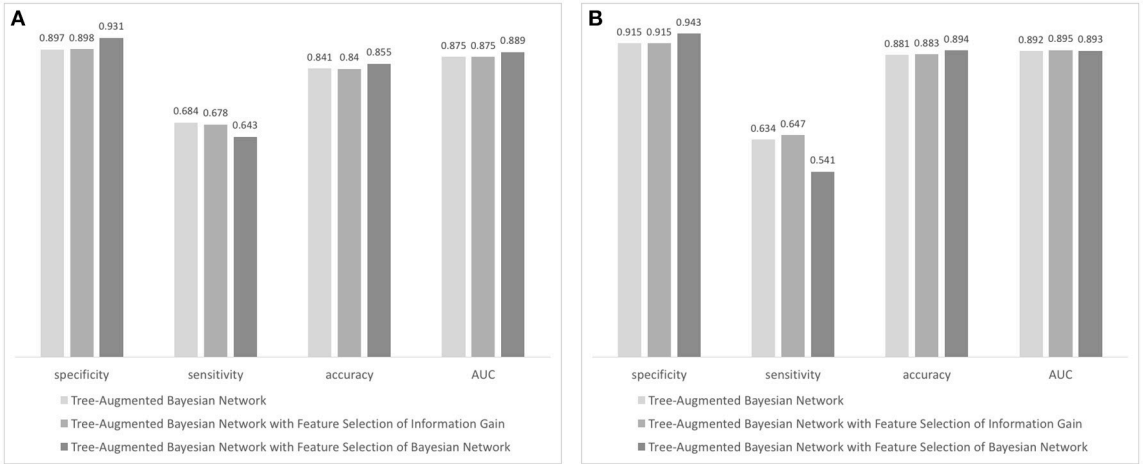


FIGURE 3 | Performance evaluation of Bayesian network-based classifiers: **(A)** performance of classifiers forecasting 90-day functional independence and **(B)** performance of classifiers for 1-year mortality prediction.

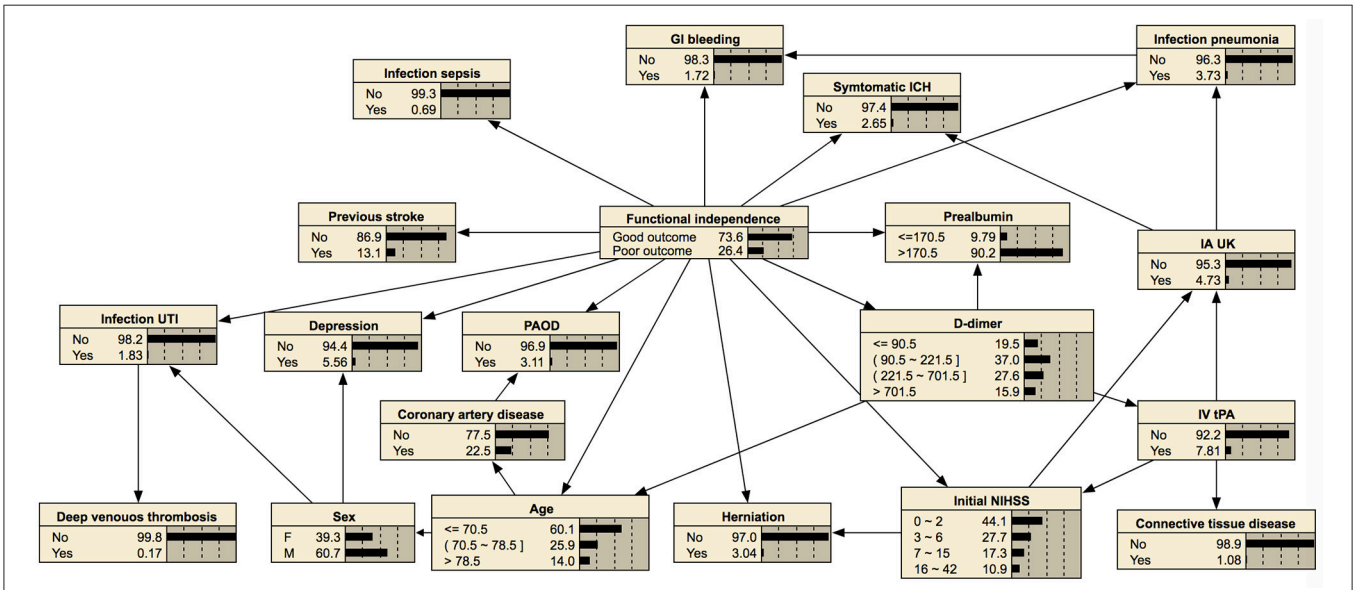


FIGURE 4 | Bayesian network for predicting functional independence at 3 months. The tree-augmented Bayesian network used 19 variables selected by the wrapper of the Bayesian network for prediction.

DISCUSSION

Interpretability is a core requirement for machine learning models in medicine, because both patients and physicians need to understand the reason behind a prediction (51). This study presents an evaluation of Bayesian networks in providing post-stroke outcomes estimates based on the collected demographic data, lab result, and initial neurological assessment. The stroke-specific variables were selected from a large stroke registry, and our experiment filtered those variables into the Bayesian network-suitable reduced set. The trained Bayesian networks were embedded in our online prediction system.

Strength of a Bayesian Network on Stroke Outcome Measurements

Research on stroke outcomes is essential for both clinical care and policy development, because approximately two-thirds of stroke survivors continue to experience functional deficits and approximately 1 of 10 patients died within 1 year (5). The prediction of post-stroke outcomes thus requires high accuracy in classification along with the understandable result that can be explained to patients. A Bayesian network can intuitively make connections between variables in medical data and provide interpretable determination in medical decision (17, 18). Therefore, Bayesian networks are well suited for

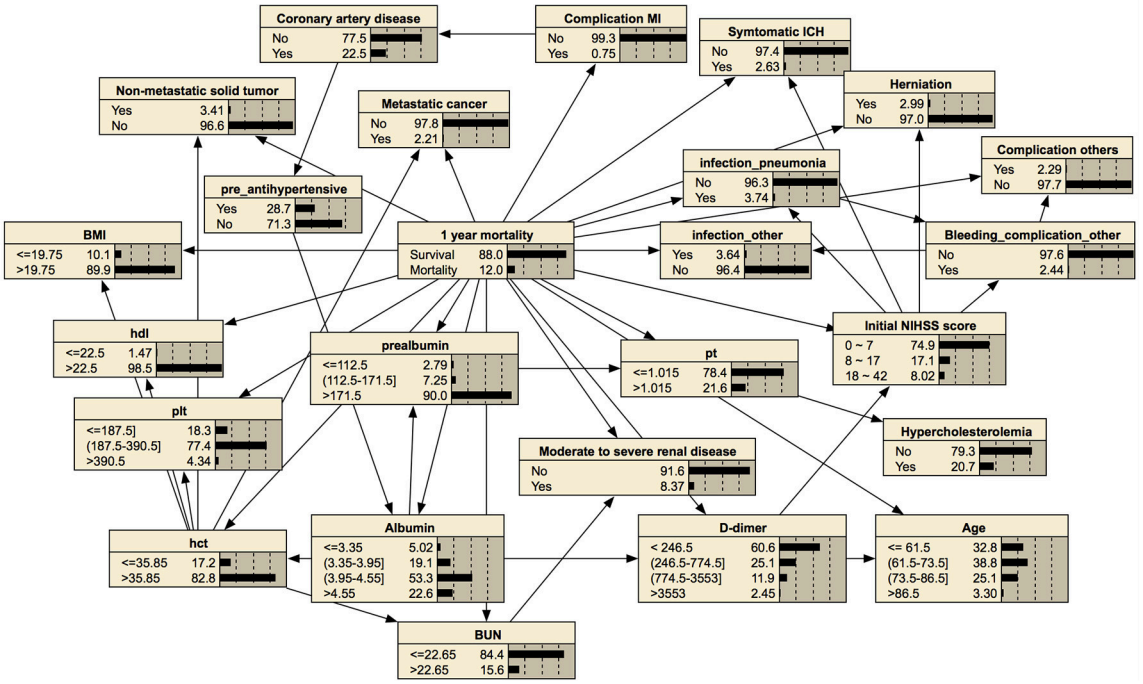


FIGURE 5 | Bayesian network for predicting 1-year mortality. The tree-augmented Bayesian network used 24 variables selected by the wrapper of the Bayesian network for prediction.

representing uncertainty and causality in prediction for patients with stroke. In recent machine learning studies, a Bayesian neural network is focused on a state of the art method which estimates predictive uncertainty (52). In Kendall and Gal (53), a Bayesian deep learning framework combines input-dependent aleatoric uncertainty together with epistemic uncertainty, to solve the black-box problem in deep learning. Constructing Bayesian networks enables medical diagnosis or prediction with incomplete and partially correct statistics, because it determines causes and effects based on the conditional probability between variables (54).

Prediction With Imbalanced Data

Often real-world data sets are predominately composed of normal instances with only a small percentage of interesting instances; therefore, class imbalance is one of the most important challenges (55). Our study also has heavily unbalanced classes in mortality prediction (3,171:434). Suppose entire positive instances were classified into negative class; then the accuracy is 0.880 in 1-year mortality prediction, although mortality is not predicted at all. Most machine learning algorithms train classifiers mainly searching for higher accuracy; therefore, the minority class is less considered in the training process. To challenge this imbalanced classification, a number of techniques have been proposed (56): oversampling approaches create minority instances by simple duplication or synthetic-minority oversampling technique (SMOTE) (57–59); certain classifiers with undersampling beat oversampling (60); cost-sensitive methods weigh higher penalty on misclassification of the minority class (61); and bagging, boosting, and hybrid approaches utilize feedback from misclassification in previous stages of learning (62).

In addition to the capability of interpretable prediction and reduced uncertainty, a Bayesian network is strong machine learning in classifying an imbalanced data set as investigated in Drummond and Holte (60) and Monsalve-Torra et al. (63). In Monsalve-Torra et al. (63), the Bayesian network outperformed radial basis function and multilayer perceptron in sensitivity. In our experiment, the learning process searched the best Bayesian network structure and parameters for the highest AUC while it guarantees at least 0.5 in sensitivity. A more computation-expensive searching algorithm such as repeated hill climbing might be helpful to increase sensitivity in classification.

Visualized Probability of Outcomes After Stroke

Bayesian networks can also provide a visual graph structure. We constructed a tree-augmented Bayesian network structure that shows an association between nodes. This visualization of conditional probability might be helpful for clinical reasoning. For example, a Bayesian network can provide the association among symptomatic intracranial hemorrhage, higher initial NIHSS score, or higher 1-year mortality with conditional probability, as shown in **Figure 5**. Therefore, our prediction model of post-stroke outcomes differs from the black-box concept of other machine learning methods (54).

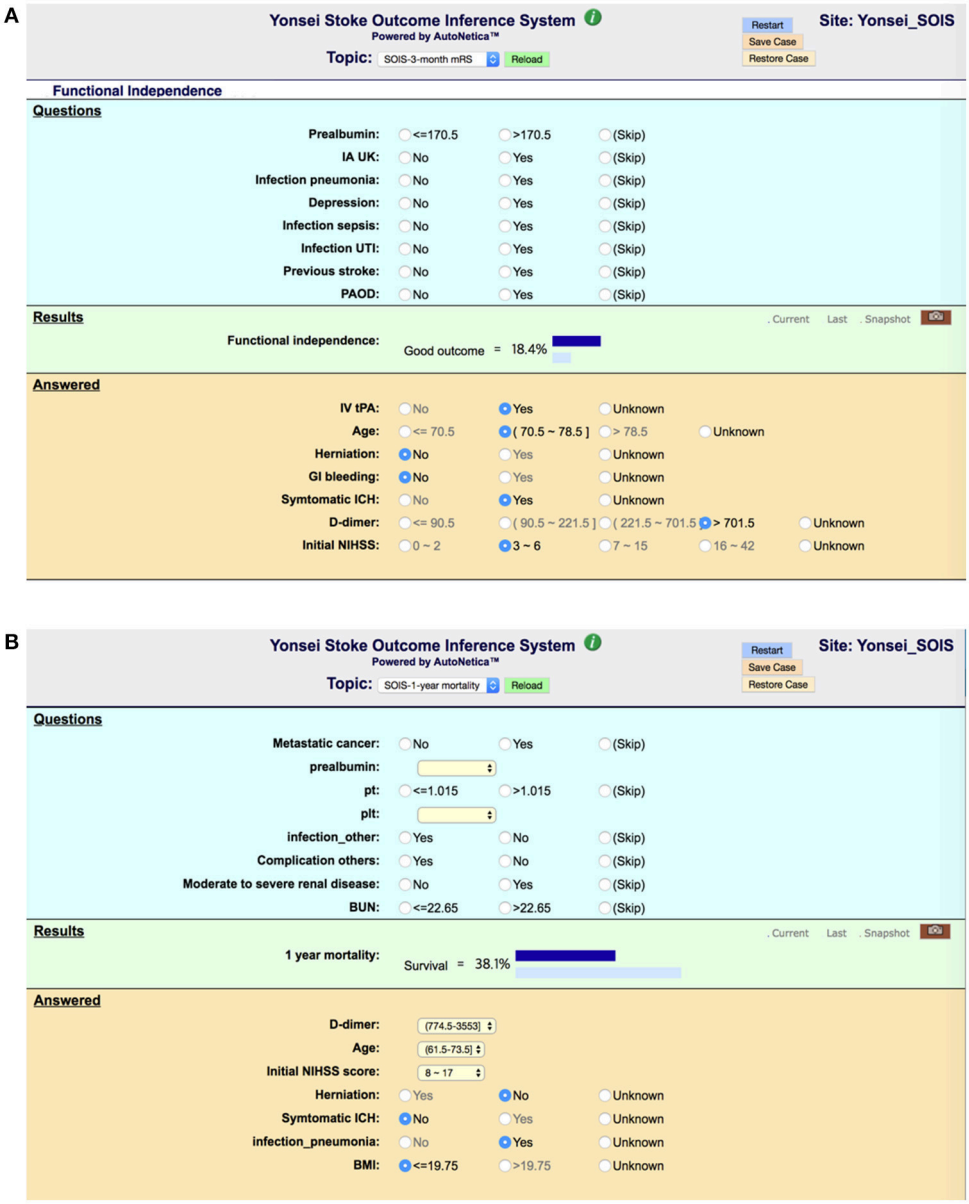


FIGURE 6 | Screenshots of an online prediction system, Y-SOIS (Yonsei Stroke Outcome Inference System). **(A)** Y-SOIS forecasts the functional independence at 3 months and **(B)** Y-SOIS forecasting the 1-year mortality.

The reduction of dimension is also helpful to visualize inference of prediction. The results demonstrated that the Bayesian network classifier with a reduced variable set can adapt the size of a network for better interpretability with a minimal or better impact on other performance.

Predictors of Post-stroke Outcomes

In this study, the information gain analysis showed that "D-dimer" was the highest feature in predicting 1-year mortality. We previously reported that a high D-dimer level by itself appeared to be associated with an increased risk of mortality (64). D-dimer can be found to be elevated in various thrombotic and inflammatory conditions, including ischemic heart disease, infection, or malignancy. These conditions are frequently found in patients with stroke and can increase the risk of mortality (65). However, patients with comorbid diseases were frequently excluded from the clinical trials, so there are no guidelines and evidence whether to treat or not patients with serious comorbid diseases in real clinical practice. In this respect, providing information of the impact of the comorbid condition with a Bayesian network might be helpful to predict the outcomes.

LIMITATIONS AND FUTURE DIRECTION

This study was conducted in a single university hospital and focused on those of East Asian descent. To provide generalizability on our prediction system, we will include various cohorts including different ethnics or patients who received thrombolysis or endovascular thrombectomy. We have plan to apply the interpretable prediction for the SECRET (SElection CRiteria in Endovascular thrombectomy and Thrombolytic therapy) study, which is a nationwide registry for hyperacute stroke. Consecutive patients who received intravenous thrombolysis and/or endovascular thrombectomy were registered (Clinical Trial Registration: NCT02964052). Bayesian network analysis of this specific condition can be used to predict outcome in patients with hyperacute stroke. We will also enlarge our training data including data of various populations by applying the proposed solution to global data archives. Additive risk predictors might be selected as determinant features in a Bayesian network, and it makes the prediction system more applicable in a global clinical environment.

AUTHOR CONTRIBUTIONS

HN designed the study; EP analyzed the data and wrote the manuscript; and H-jC and HN contributed to data interpretation and revising the manuscript.

FUNDING

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B03029014) and grant funded by the Korea government (MSIP) (2016R1C1B2016028) and the National Fire Agency, Republic of Korea (MPSS-2015-70).

REFERENCES

1. Ko Y, Park JH, Kim WJ, Yang MH, Kwon OK, Oh CW, et al. The long-term incidence of recurrent stroke: single hospital-based cohort study. *J Korean Neurol Assoc*. (2009) 27:110–5.
2. Albers GW, Marks MP, Kemp S, Christensen S, Tsai JP, Ortega-Gutierrez S, et al. Thrombectomy for stroke at 6 to 16 hours with selection by perfusion imaging. *New Engl J Med*. (2018) 378:708–18. doi: 10.1056/NEJMoa1713973
3. Anderson CS, Woodward M, Chalmers J. More on low-dose versus standard-dose intravenous alteplase in acute ischemic stroke. *New Engl J Med*. (2018) 378:1465–6. doi: 10.1056/NEJMc1801548
4. Banks JL, Marotta CA. Outcomes validity and reliability of the modified Rankin scale: implications for stroke clinical trials: a literature review and synthesis. *Stroke* (2007) 38:1091–6. doi: 10.1161/01.STR.0000258355.23810.c6
5. Nam HS, Kim HC, Kim YD, Lee HS, Kim J, Lee DH, et al. Long-term mortality in patients with stroke of undetermined etiology. *Stroke* (2012) 43:2948–56. doi: 10.1161/STROKEAHA.112.661074
6. Cruz JA, Wishart DS. Applications of machine learning in cancer prediction and prognosis. *Cancer Inf*. (2006) 2:117693510600200030. doi: 10.1177/117693510600200030
7. Obermeyer Z, Emanuel EJ. Predicting the future—big data, machine learning, and clinical medicine. *New Engl J Med*. (2016) 375:1216. doi: 10.1056/NEJMp1606181
8. Asadi H, Dowling R, Yan B, Mitchell P. Machine learning for outcome prediction of acute ischemic stroke post intra-arterial therapy. *PLoS ONE* (2014) 9:e88225. doi: 10.1371/journal.pone.0088225
9. Magnin B, Mesrob L, Kinkingnéhun S, Pélégrini-Issac M, Colliot O, Sarazin M, et al. Support vector machine-based classification of Alzheimer's disease from whole-brain anatomical MRI. *Neuroradiology* (2009) 51:73–83. doi: 10.1007/s00234-008-0463-x
10. Tjortjis C, Saraee M, Theodoulidis B, Keane J. Using T3, an improved decision tree classifier, for mining stroke-related medical data. *Methods Inf Med*. (2007) 46:523–9. doi: 10.1160/ME0317
11. Ward MM, Pajevic S, Dreyfuss J, Malley JD. Short-term prediction of mortality in patients with systemic lupus erythematosus: classification of outcomes using random forests. *Arthritis Care Res*. (2006) 55:74–80. doi: 10.1002/art.21695
12. Statnikov A, Wang L, Aliferis CF. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinform*. (2008) 9:319. doi: 10.1186/1471-2105-9-319
13. Witten IH, Frank E, Hall MA, Pal CJ. *Data Mining: Practical Machine Learning Tools and Techniques*. Burlington, MA: Morgan Kaufmann (2016).
14. Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. *Mach Learn*. (1997) 29:131–63. doi: 10.1023/A:1007465528199
15. Lucas PJ, Van der Gaag LC, Abu-Hanna A. Bayesian networks in biomedicine and health-care. *Artif Intell Med*. (2004) 30:201–14. doi: 10.1016/j.artmed.2003.11.001
16. Nikovski D. Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics. *IEEE Trans Knowl Data Eng*. (2000) 12:509–16. doi: 10.1109/69.868904
17. Letham B, Rudin C, McCormick TH, Madigan D. Interpretable classifiers using rules and bayesian analysis: building a better stroke prediction model. *Ann Appl Stat*. (2015) 9:1350–71. doi: 10.1214/15-AOAS848
18. Jansen R, Yu H, Greenbaum D, Kluger Y, Krogan NJ, Chung S, et al. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* (2003) 302:449–53. doi: 10.1126/science.1087361
19. Heckerman D, Geiger D, Chickering DM. Learning bayesian networks: the combination of knowledge and statistical data. *Mach Learn*. (1995) 20:197–243. doi: 10.1007/BF00994016
20. Uusitalo L. Advantages and challenges of Bayesian networks in environmental modelling. *Ecol Model*. (2007) 203:312–8. doi: 10.1016/j.ecolmodel.2006.11.033
21. Kononenko I. Machine learning for medical diagnosis: history, state of the art and perspective. *Artif Intell Med*. (2001) 23:89–109. doi: 10.1016/S0933-3657(01)00077-X
22. Sesen MB, Nicholson AE, Banares-Alcantara R, Kadir T, Brady M. Bayesian networks for clinical decision support in lung cancer care. *PLoS ONE* (2013) 8:e82349. doi: 10.1371/journal.pone.0082349
23. Lee BI, Nam HS, Heo JH, Kim DI. Yonsei stroke registry. *Cerebrovasc Dis*. (2001) 12:145–51. doi: 10.1159/000047697
24. Cho HJ, Choi HY, Kim YD, Nam HS, Han SW, Ha JW, et al. Transoesophageal echocardiography in patients with acute stroke with sinus rhythm and no cardiac disease history. *J Neurol Neurosurg Psychiatry* (2010) 81:412–5. doi: 10.1136/jnnp.2009.190322
25. Yoo J, Yang JH, Choi BW, Kim YD, Nam HS, Choi HY, et al. The frequency and risk of preclinical coronary artery disease detected using multichannel cardiac computed tomography in patients with ischemic stroke. *Cerebrovasc Dis*. (2012) 33:286–94. doi: 10.1159/000334980
26. Song TJ, Kim J, Lee HS, Nam CM, Nam HS, Kim YD, et al. Distribution of cerebral microbleeds determines their association with impaired kidney function. *J Clin Neurol*. (2014) 10:222–8. doi: 10.3988/jcn.2014.10.3.222
27. Adams HPJr, Bendixen BH, Kappelle LJ, Biller J, Love BB, Gordon DL, et al. Classification of subtype of acute ischemic stroke. Definitions for use in a multicenter clinical trial. TOAST. Trial of Org 10172 in Acute Stroke Treatment. *Stroke* (1993) 24:35–41. doi: 10.1161/01.STR.24.1.35

28. Khang Y-H, Lynch JW, Kaplan GA. Health inequalities in Korea: age-and sex-specific educational differences in the 10 leading causes of death. *Int J Epidemiol*. (2004) 33:299–308. doi: 10.1093/ije/dyg244
29. Kim HC, Choi DP, Ahn SV, Nam CM, Suh I. Six-year survival and causes of death among stroke patients in Korea. *Neuroepidemiology* (2009) 32:94–100. doi: 10.1159/000177034
30. Drugan MM, Wiering MA. Feature selection for Bayesian network classifiers using the MDL-FS score. *Int J Approx Reason*. (2010) 51:695–717. doi: 10.1016/j.ijar.2010.02.001
31. Liu Z, Malone B, Yuan C. Empirical evaluation of scoring functions for Bayesian network model selection. *BMC Bioinformatics* (2012) 13(Suppl. 15):S14. doi: 10.1186/1471-2105-13-S15-S14
32. Schwarz G. Estimating the dimension of a model. *Ann Stat*. (1978) 6:461–4. doi: 10.1214/aos/1176344136
33. Vrieze SI. Model selection and psychological theory: a discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychol Methods* (2012) 17:228. doi: 10.1037/a0027127
34. Lam W, Bacchus F. Learning Bayesian belief networks: an approach based on the MDL principle. *Comput Intell*. (1994) 10:269–93. doi: 10.1111/j.1467-8640.1994.tb00166.x
35. Tsamardinos I, Brown LE, Aliferis CF. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach Learn*. (2006) 65:31–78. doi: 10.1007/s10994-006-6889-7
36. Chinnasamy A, Sung W-K, Mittal A. Protein structure and fold prediction using tree-augmented naive Bayesian classifier. *J Bioinform Comput Biol*. (2005) 3:803–19. doi: 10.1142/S0219720005001302
37. Guyon I, Elisseeff A. An introduction to variable and feature selection. *J Mach Learn Res*. (2003) 3:1157–82.
38. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. science. (2000) 290:2323–6. doi: 10.1126/science.290.5500.2323
39. Hruschka ER, Hruschka ER, Ebecken NF. Feature selection by Bayesian networks. In: *Conference of the Canadian Society for Computational Studies of Intelligence*. London, ON: Springer (2004).
40. Jović A, Brkić K, Bogunović N. A review of feature selection methods with applications. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* Opatija (2015).
41. Lee C, Lee GG. Information gain and divergence-based feature selection for machine learning-based text categorization. *Informat Proc Manag*. (2006) 42:155–65. doi: 10.1016/j.ipm.2004.08.006
42. Motwani M, Dey D, Berman DS, Germano G, Achenbach S, Al-Mallah MH, et al. Machine learning for prediction of all-cause mortality in patients with suspected coronary artery disease: a 5-year multicentre prospective registry analysis. *Eur Heart J*. (2016) 38:500–7. doi: 10.1093/eurheartj/ehw188
43. Lei S. A feature selection method based on information gain and genetic algorithm. In: *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on* Hangzhou: IEEE (2012).
44. Kononenko I. Estimating attributes: analysis and extensions of RELIEF. In: *European Conference on Machine Learning*. Catania: Springer (1994).
45. Robnik-Šikonja M, Kononenko I. An adaptation of Relief for attribute estimation in regression. In: *Proceedings of the 14th International Conference on Machine Learning (ICML)*. San Francisco, CA (1997).
46. Hall MA. Correlation-based feature selection of discrete and numeric class machine learning.In: *ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco, CA (2000).
47. Yu L, Liu H. Feature selection for high-dimensional data: a fast correlation-based filter solution. In: *Proceedings of the 20th International Conference on Machine Learning (ICML)*. Washington, DC (2003).
48. Bermejo P, Gamez JA, Puerta JM. Improving incremental wrapper-based subset selection via replacement and early stopping. *Int J Pattern Recogn Artif Intell*. (2011) 25:605–25. doi: 10.1142/S0218001411008804
49. Kohavi R, John GH. Wrappers for feature subset selection. *Artif Intell*. (1997) 97:273–324. doi: 10.1016/S0004-3702(97)00043-X
50. Arlot S, Celisse A. A survey of cross-validation procedures for model selection. *Stat Surv*. (2010) 4:40–79. doi: 10.1214/09-SS054
51. Valdes G, Luna JM, Eaton E, Simone II CB, Ungar LH, Solberg TD. MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine. *Sci Rep*. (2016) 6:37854. doi: 10.1038/srep37854
52. Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. In: *Advances in Neural Information Processing Systems*. Long Beach, CA (2017).
53. Kendall A, Gal Y. What uncertainties do we need in bayesian deep learning for computer vision? In: *Advances in Neural Information Processing Systems*. Long Beach, CA (2017).
54. Korb KB, Nicholson AE. *Bayesian Artificial Intelligence*. Boca Raton, FL: CRC Press (2010).
55. Maldonado S, Weber R, Famili F. Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. *Inf Sci*. (2014) 286:228–46. doi: 10.1016/j.ins.2014.07.015
56. Fernández A, García S, Herrera F. Addressing the classification with imbalanced data: open problems and new challenges on class distribution. In: *International Conference on Hybrid Artificial Intelligence Systems*. Wroclaw: Springer (2011).
57. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res*. (2002) 16:321–57. doi: 10.1613/jair.953
58. Han H, Wang W-Y, Mao B-H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: *International Conference on Intelligent Computing*. Hefei: Springer (2005).
59. Ho KC, Speier W, El-Saden S, Liebeskind DS, Saver JL, Bui AA, et al. Predicting discharge mortality after acute ischemic stroke using balanced data. In: *AMIA Annual Symposium Proceedings, American Medical Informatics Association*. Washington DC (2014).
60. Drummond C, Holte RC. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: *Workshop on Learning from Imbalanced Datasets II*. Washington DC: Citeseer (2003).
61. Weiss GM, McCarthy K, Zabar B. Cost-sensitive learning vs. sampling: which is best for handling unbalanced classes with unequal error costs? In: *IEEE International Conference on Data Mining* (2007) p. 35–41.
62. Manual N. *Netica V5. 18*. Vancouver, BC: Norsys Software Corp. (2015).
63. Monsalve-Torra A, Ruiz-Fernandez D, Marin-Alonso O, Soriano-Payá A, Camacho-Mackenzie J, Carreño-Jaimes M. Using machine learning methods for predicting inhospital mortality in patients undergoing open repair of abdominal aortic aneurysm. *J Biomed Inf*. (2016) 62:195–201. doi: 10.1016/j.jbi.2016.07.007
64. Kim YD, Song D, Nam HS, Lee K, Yoo J, Hong G-R, et al. D-dimer for prediction of long-term outcome in cryptogenic stroke patients with patent foramen ovale. *Thromb Haemost*. (2015) 114:614–22. doi: 10.1160/TH14-12-1040
65. Adam SS, Key NS, Greenberg CS. D-dimer antigen: current concepts and future prospects. *Blood* (2009) 113:2878–87. doi: 10.1182/blood-2008-06-165845

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.