{

"cells": [

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"# Credit Card Fraud Detection - Sagnik Ghosh"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"## Data frames"

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {},

"outputs": [],

"source": [

"import pandas as pd"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"metadata": {},

"outputs": [],

"source": [

"df = pd.read\_csv('creditcard.csv')"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Time</th>\n",

" <th>V1</th>\n",

" <th>V2</th>\n",

" <th>V3</th>\n",

" <th>V4</th>\n",

" <th>V5</th>\n",

" <th>V6</th>\n",

" <th>V7</th>\n",

" <th>V8</th>\n",

" <th>V9</th>\n",

" <th>...</th>\n",

" <th>V21</th>\n",

" <th>V22</th>\n",

" <th>V23</th>\n",

" <th>V24</th>\n",

" <th>V25</th>\n",

" <th>V26</th>\n",

" <th>V27</th>\n",

" <th>V28</th>\n",

" <th>Amount</th>\n",

" <th>Class</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>0.0</td>\n",

" <td>-1.359807</td>\n",

" <td>-0.072781</td>\n",

" <td>2.536347</td>\n",

" <td>1.378155</td>\n",

" <td>-0.338321</td>\n",

" <td>0.462388</td>\n",

" <td>0.239599</td>\n",

" <td>0.098698</td>\n",

" <td>0.363787</td>\n",

" <td>...</td>\n",

" <td>-0.018307</td>\n",

" <td>0.277838</td>\n",

" <td>-0.110474</td>\n",

" <td>0.066928</td>\n",

" <td>0.128539</td>\n",

" <td>-0.189115</td>\n",

" <td>0.133558</td>\n",

" <td>-0.021053</td>\n",

" <td>149.62</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>0.0</td>\n",

" <td>1.191857</td>\n",

" <td>0.266151</td>\n",

" <td>0.166480</td>\n",

" <td>0.448154</td>\n",

" <td>0.060018</td>\n",

" <td>-0.082361</td>\n",

" <td>-0.078803</td>\n",

" <td>0.085102</td>\n",

" <td>-0.255425</td>\n",

" <td>...</td>\n",

" <td>-0.225775</td>\n",

" <td>-0.638672</td>\n",

" <td>0.101288</td>\n",

" <td>-0.339846</td>\n",

" <td>0.167170</td>\n",

" <td>0.125895</td>\n",

" <td>-0.008983</td>\n",

" <td>0.014724</td>\n",

" <td>2.69</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>1.0</td>\n",

" <td>-1.358354</td>\n",

" <td>-1.340163</td>\n",

" <td>1.773209</td>\n",

" <td>0.379780</td>\n",

" <td>-0.503198</td>\n",

" <td>1.800499</td>\n",

" <td>0.791461</td>\n",

" <td>0.247676</td>\n",

" <td>-1.514654</td>\n",

" <td>...</td>\n",

" <td>0.247998</td>\n",

" <td>0.771679</td>\n",

" <td>0.909412</td>\n",

" <td>-0.689281</td>\n",

" <td>-0.327642</td>\n",

" <td>-0.139097</td>\n",

" <td>-0.055353</td>\n",

" <td>-0.059752</td>\n",

" <td>378.66</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>1.0</td>\n",

" <td>-0.966272</td>\n",

" <td>-0.185226</td>\n",

" <td>1.792993</td>\n",

" <td>-0.863291</td>\n",

" <td>-0.010309</td>\n",

" <td>1.247203</td>\n",

" <td>0.237609</td>\n",

" <td>0.377436</td>\n",

" <td>-1.387024</td>\n",

" <td>...</td>\n",

" <td>-0.108300</td>\n",

" <td>0.005274</td>\n",

" <td>-0.190321</td>\n",

" <td>-1.175575</td>\n",

" <td>0.647376</td>\n",

" <td>-0.221929</td>\n",

" <td>0.062723</td>\n",

" <td>0.061458</td>\n",

" <td>123.50</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2.0</td>\n",

" <td>-1.158233</td>\n",

" <td>0.877737</td>\n",

" <td>1.548718</td>\n",

" <td>0.403034</td>\n",

" <td>-0.407193</td>\n",

" <td>0.095921</td>\n",

" <td>0.592941</td>\n",

" <td>-0.270533</td>\n",

" <td>0.817739</td>\n",

" <td>...</td>\n",

" <td>-0.009431</td>\n",

" <td>0.798278</td>\n",

" <td>-0.137458</td>\n",

" <td>0.141267</td>\n",

" <td>-0.206010</td>\n",

" <td>0.502292</td>\n",

" <td>0.219422</td>\n",

" <td>0.215153</td>\n",

" <td>69.99</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>284802</th>\n",

" <td>172786.0</td>\n",

" <td>-11.881118</td>\n",

" <td>10.071785</td>\n",

" <td>-9.834783</td>\n",

" <td>-2.066656</td>\n",

" <td>-5.364473</td>\n",

" <td>-2.606837</td>\n",

" <td>-4.918215</td>\n",

" <td>7.305334</td>\n",

" <td>1.914428</td>\n",

" <td>...</td>\n",

" <td>0.213454</td>\n",

" <td>0.111864</td>\n",

" <td>1.014480</td>\n",

" <td>-0.509348</td>\n",

" <td>1.436807</td>\n",

" <td>0.250034</td>\n",

" <td>0.943651</td>\n",

" <td>0.823731</td>\n",

" <td>0.77</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>284803</th>\n",

" <td>172787.0</td>\n",

" <td>-0.732789</td>\n",

" <td>-0.055080</td>\n",

" <td>2.035030</td>\n",

" <td>-0.738589</td>\n",

" <td>0.868229</td>\n",

" <td>1.058415</td>\n",

" <td>0.024330</td>\n",

" <td>0.294869</td>\n",

" <td>0.584800</td>\n",

" <td>...</td>\n",

" <td>0.214205</td>\n",

" <td>0.924384</td>\n",

" <td>0.012463</td>\n",

" <td>-1.016226</td>\n",

" <td>-0.606624</td>\n",

" <td>-0.395255</td>\n",

" <td>0.068472</td>\n",

" <td>-0.053527</td>\n",

" <td>24.79</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>284804</th>\n",

" <td>172788.0</td>\n",

" <td>1.919565</td>\n",

" <td>-0.301254</td>\n",

" <td>-3.249640</td>\n",

" <td>-0.557828</td>\n",

" <td>2.630515</td>\n",

" <td>3.031260</td>\n",

" <td>-0.296827</td>\n",

" <td>0.708417</td>\n",

" <td>0.432454</td>\n",

" <td>...</td>\n",

" <td>0.232045</td>\n",

" <td>0.578229</td>\n",

" <td>-0.037501</td>\n",

" <td>0.640134</td>\n",

" <td>0.265745</td>\n",

" <td>-0.087371</td>\n",

" <td>0.004455</td>\n",

" <td>-0.026561</td>\n",

" <td>67.88</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>284805</th>\n",

" <td>172788.0</td>\n",

" <td>-0.240440</td>\n",

" <td>0.530483</td>\n",

" <td>0.702510</td>\n",

" <td>0.689799</td>\n",

" <td>-0.377961</td>\n",

" <td>0.623708</td>\n",

" <td>-0.686180</td>\n",

" <td>0.679145</td>\n",

" <td>0.392087</td>\n",

" <td>...</td>\n",

" <td>0.265245</td>\n",

" <td>0.800049</td>\n",

" <td>-0.163298</td>\n",

" <td>0.123205</td>\n",

" <td>-0.569159</td>\n",

" <td>0.546668</td>\n",

" <td>0.108821</td>\n",

" <td>0.104533</td>\n",

" <td>10.00</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>284806</th>\n",

" <td>172792.0</td>\n",

" <td>-0.533413</td>\n",

" <td>-0.189733</td>\n",

" <td>0.703337</td>\n",

" <td>-0.506271</td>\n",

" <td>-0.012546</td>\n",

" <td>-0.649617</td>\n",

" <td>1.577006</td>\n",

" <td>-0.414650</td>\n",

" <td>0.486180</td>\n",

" <td>...</td>\n",

" <td>0.261057</td>\n",

" <td>0.643078</td>\n",

" <td>0.376777</td>\n",

" <td>0.008797</td>\n",

" <td>-0.473649</td>\n",

" <td>-0.818267</td>\n",

" <td>-0.002415</td>\n",

" <td>0.013649</td>\n",

" <td>217.00</td>\n",

" <td>0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>284807 rows × 31 columns</p>\n",

"</div>"

],

"text/plain": [

" Time V1 V2 V3 V4 V5 \\\n",

"0 0.0 -1.359807 -0.072781 2.536347 1.378155 -0.338321 \n",

"1 0.0 1.191857 0.266151 0.166480 0.448154 0.060018 \n",

"2 1.0 -1.358354 -1.340163 1.773209 0.379780 -0.503198 \n",

"3 1.0 -0.966272 -0.185226 1.792993 -0.863291 -0.010309 \n",

"4 2.0 -1.158233 0.877737 1.548718 0.403034 -0.407193 \n",

"... ... ... ... ... ... ... \n",

"284802 172786.0 -11.881118 10.071785 -9.834783 -2.066656 -5.364473 \n",

"284803 172787.0 -0.732789 -0.055080 2.035030 -0.738589 0.868229 \n",

"284804 172788.0 1.919565 -0.301254 -3.249640 -0.557828 2.630515 \n",

"284805 172788.0 -0.240440 0.530483 0.702510 0.689799 -0.377961 \n",

"284806 172792.0 -0.533413 -0.189733 0.703337 -0.506271 -0.012546 \n",

"\n",

" V6 V7 V8 V9 ... V21 V22 \\\n",

"0 0.462388 0.239599 0.098698 0.363787 ... -0.018307 0.277838 \n",

"1 -0.082361 -0.078803 0.085102 -0.255425 ... -0.225775 -0.638672 \n",

"2 1.800499 0.791461 0.247676 -1.514654 ... 0.247998 0.771679 \n",

"3 1.247203 0.237609 0.377436 -1.387024 ... -0.108300 0.005274 \n",

"4 0.095921 0.592941 -0.270533 0.817739 ... -0.009431 0.798278 \n",

"... ... ... ... ... ... ... ... \n",

"284802 -2.606837 -4.918215 7.305334 1.914428 ... 0.213454 0.111864 \n",

"284803 1.058415 0.024330 0.294869 0.584800 ... 0.214205 0.924384 \n",

"284804 3.031260 -0.296827 0.708417 0.432454 ... 0.232045 0.578229 \n",

"284805 0.623708 -0.686180 0.679145 0.392087 ... 0.265245 0.800049 \n",

"284806 -0.649617 1.577006 -0.414650 0.486180 ... 0.261057 0.643078 \n",

"\n",

" V23 V24 V25 V26 V27 V28 Amount \\\n",

"0 -0.110474 0.066928 0.128539 -0.189115 0.133558 -0.021053 149.62 \n",

"1 0.101288 -0.339846 0.167170 0.125895 -0.008983 0.014724 2.69 \n",

"2 0.909412 -0.689281 -0.327642 -0.139097 -0.055353 -0.059752 378.66 \n",

"3 -0.190321 -1.175575 0.647376 -0.221929 0.062723 0.061458 123.50 \n",

"4 -0.137458 0.141267 -0.206010 0.502292 0.219422 0.215153 69.99 \n",

"... ... ... ... ... ... ... ... \n",

"284802 1.014480 -0.509348 1.436807 0.250034 0.943651 0.823731 0.77 \n",

"284803 0.012463 -1.016226 -0.606624 -0.395255 0.068472 -0.053527 24.79 \n",

"284804 -0.037501 0.640134 0.265745 -0.087371 0.004455 -0.026561 67.88 \n",

"284805 -0.163298 0.123205 -0.569159 0.546668 0.108821 0.104533 10.00 \n",

"284806 0.376777 0.008797 -0.473649 -0.818267 -0.002415 0.013649 217.00 \n",

"\n",

" Class \n",

"0 0 \n",

"1 0 \n",

"2 0 \n",

"3 0 \n",

"4 0 \n",

"... ... \n",

"284802 0 \n",

"284803 0 \n",

"284804 0 \n",

"284805 0 \n",

"284806 0 \n",

"\n",

"[284807 rows x 31 columns]"

]

},

"execution\_count": 3,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"0 284315\n",

"1 492\n",

"Name: Class, dtype: int64"

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df['Class'].value\_counts()"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"### Data Pre-processing"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"metadata": {},

"outputs": [],

"source": [

"from sklearn.model\_selection import train\_test\_split"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"metadata": {},

"outputs": [],

"source": [

"from sklearn.preprocessing import StandardScaler"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"metadata": {},

"outputs": [],

"source": [

"scalar = StandardScaler()"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"metadata": {},

"outputs": [],

"source": [

"X = df.drop('Class', axis=1)\n",

"y = df.Class"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"metadata": {},

"outputs": [],

"source": [

"X = scalar.fit\_transform(X)"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"metadata": {},

"outputs": [],

"source": [

"X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"## Modeling"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"metadata": {},

"outputs": [],

"source": [

"from sklearn.svm import SVC"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"metadata": {},

"outputs": [],

"source": [

"model\_svc = SVC()"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"SVC()"

]

},

"execution\_count": 13,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model\_svc.fit(X\_train, y\_train)"

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"0.9996752178015756"

]

},

"execution\_count": 14,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model\_svc.score(X\_train,y\_train)"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"0.999385555282469"

]

},

"execution\_count": 15,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model\_svc.score(X\_test,y\_test)"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"metadata": {},

"outputs": [],

"source": [

"y\_predict = model\_svc.predict(X\_test)"

]

},

{

"cell\_type": "markdown",

"metadata": {},

"source": [

"## Implementing Report"

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"metadata": {},

"outputs": [],

"source": [

"from sklearn.metrics import classification\_report , confusion\_matrix"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"metadata": {},

"outputs": [],

"source": [

"import numpy as np"

]

},

{

"cell\_type": "code",

"execution\_count": 19,

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>predicted fraud</th>\n",

" <th>predicted normal</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>is Fraud</th>\n",

" <td>55</td>\n",

" <td>32</td>\n",

" </tr>\n",

" <tr>\n",

" <th>is Normal</th>\n",

" <td>3</td>\n",

" <td>56872</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" predicted fraud predicted normal\n",

"is Fraud 55 32\n",

"is Normal 3 56872"

]

},

"execution\_count": 19,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"cm = np.array(confusion\_matrix(y\_test, y\_predict, labels=[1,0]))\n",

"confusion = pd.DataFrame(cm, index=['is Fraud', 'is Normal'],columns=['predicted fraud','predicted normal'])\n",

"confusion"

]

},

{

"cell\_type": "code",

"execution\_count": 20,

"metadata": {},

"outputs": [],

"source": [

"import seaborn as sns"

]

},

{

"cell\_type": "code",

"execution\_count": 21,

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.axes.\_subplots.AxesSubplot at 0x1a0374fe948>"

]

},

"execution\_count": 21,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png":

"text/plain": [

"<Figure size 432x288 with 2 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"sns.heatmap(confusion, annot=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" precision recall f1-score support\n",

"\n",

" 0 1.00 1.00 1.00 56875\n",

" 1 0.95 0.63 0.76 87\n",

"\n",

" accuracy 1.00 56962\n",

" macro avg 0.97 0.82 0.88 56962\n",

"weighted avg 1.00 1.00 1.00 56962\n",

"\n"

]

}

],

"source": [

"print(classification\_report(y\_test, y\_predict))"

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.7.7"

}

},

"nbformat": 4,

"nbformat\_minor": 4

}