NB_dkCashLineup.ipynb (page 1)


```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "36e2251b",
   "metadata": {},
   "outputs": [],
   "source": [
    "from utils.draftkingsCashOptimizer import fetch_player_data, normalize_features,
generate_top_lineups"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "4f813639",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "application/vnd.microsoft.datawrangler.viewer.v0+json": {
       "columns": [
        {
         "name": "index",
         "rawType": "int64",
         "type": "integer"
        },
        {
         "name": "id",
         "rawType": "int64",
         "type": "integer"
        },
        {
         "name": "name",
         "rawType": "object",
         "type": "string"
        },
        {
         "name": "position",
         "rawType": "object",
         "type": "string"
        },
        {
         "name": "roster position",
         "rawType": "object",
         "type": "string"
        },
        {
         "name": "salary",
         "rawType": "int64",
         "type": "integer"
        },
```

```
 {
  "name": "game info",
  "rawType": "object",
  "type": "string"
 },
 {
  "name": "team",
  "rawType": "object",
  "type": "string"
 },
 {
  "name": "avgpointspergame",
  "rawType": "float64",
  "type": "float"
 },
 {
  "name": "projpts",
  "rawType": "float64",
  "type": "float"
 },
 {
  "name": "projown",
  "rawType": "float64",
  "type": "float"
 }
],
"ref": "64c59038-d6ce-430f-90e5-855936bd289d",
"rows": [
 [
  "0",
  "40142470",
  "Christian McCaffrey",
  "RB",
  "RB/FLEX",
  "8500",
  "JAX@SF 09/28/2025 04:05PM ET",
  "49ers",
  "23.3",
  "23.5",
  "32.0"
 ],
 [
  "1",
  "40142472",
  "Bijan Robinson",
  "RB",
  "RB/FLEX",
  "8200",
  "WAS@ATL 09/28/2025 01:00PM ET",
  "Falcons",
  "22.1",
  "22.0",
```

```
   "15.8"
  ],
  [
   "2",
   "40142716",
   "Puka Nacua",
   "WR",
   "WR/FLEX",
   "7900",
   "IND@LAR 09/28/2025 04:05PM ET",
   "Rams",
   "26.5",
   "20.0",
   "38.4"
  ]
 ],
 "shape": {
  "columns": 10,
  "rows": 3
 }
},
"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>id</th>\n",
 "      <th>name</th>\n",
 "      <th>position</th>\n",
 "      <th>roster position</th>\n",
 "      <th>salary</th>\n",
 "      <th>game info</th>\n",
 "      <th>team</th>\n",
 "      <th>avgpointspergame</th>\n",
 "      <th>projpts</th>\n",
 "      <th>projown</th>\n",
 "    </tr>\n",
 "  </thead>\n",
```

```
 "  <tbody>\n",
 "    <tr>\n",
 "      <th>0</th>\n",
 "      <td>40142470</td>\n",
 "      <td>Christian McCaffrey</td>\n",
 "      <td>RB</td>\n",
 "      <td>RB/FLEX</td>\n",
 "      <td>8500</td>\n",
 "      <td>JAX@SF 09/28/2025 04:05PM ET</td>\n",
 "      <td>49ers</td>\n",
 "      <td>23.3</td>\n",
 "      <td>23.5</td>\n",
 "      <td>32.0</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>1</th>\n",
 "      <td>40142472</td>\n",
 "      <td>Bijan Robinson</td>\n",
 "      <td>RB</td>\n",
 "      <td>RB/FLEX</td>\n",
 "      <td>8200</td>\n",
 "      <td>WAS@ATL 09/28/2025 01:00PM ET</td>\n",
 "      <td>Falcons</td>\n",
 "      <td>22.1</td>\n",
 "      <td>22.0</td>\n",
 "      <td>15.8</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>2</th>\n",
 "      <td>40142716</td>\n",
 "      <td>Puka Nacua</td>\n",
 "      <td>WR</td>\n",
 "      <td>WR/FLEX</td>\n",
 "      <td>7900</td>\n",
 "      <td>IND@LAR 09/28/2025 04:05PM ET</td>\n",
 "      <td>Rams</td>\n",
 "      <td>26.5</td>\n",
 "      <td>20.0</td>\n",
 "      <td>38.4</td>\n",
 "    </tr>\n",
 "  </tbody>\n",
 "</table>\n",
 "</div>"
],
"text/plain": [
 "         id                name position roster position  salary  \\\n",
 "0  40142470  Christian McCaffrey       RB         RB/FLEX    8500   \n",
 "1  40142472       Bijan Robinson       RB         RB/FLEX    8200   \n",
 "2  40142716           Puka Nacua       WR         WR/FLEX    7900   \n",
 "\n",
 "                       game info     team  avgpointspergame  projpts  projown  \n",
 "0   JAX@SF 09/28/2025 04:05PM ET    49ers              23.3     23.5     32.0  \n",
```

```
      "1  WAS@ATL 09/28/2025 01:00PM ET  Falcons              22.1     22.0     15.8  \n",
      "2  IND@LAR 09/28/2025 04:05PM ET     Rams              26.5     20.0     38.4  "
     ]
    },
    "execution_count": 2,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "player_data = fetch_player_data(\n",
   "    week=\"week04\",\n",
   "    dk_salaries_file=\"DKSalaries.csv\",\n",
   "    dk_projections_file=\"fantasy_footballers-nfl-dk-Main-projections.csv\"\n",
   "    )\n",
   "player_data.head(3)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 3,
  "id": "aa79b789",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "application/vnd.microsoft.datawrangler.viewer.v0+json": {
      "columns": [
       {
        "name": "index",
        "rawType": "int64",
        "type": "integer"
       },
       {
        "name": "id",
        "rawType": "int64",
        "type": "integer"
       },
       {
        "name": "name",
        "rawType": "object",
        "type": "string"
       },
       {
        "name": "position",
        "rawType": "object",
        "type": "string"
       },
       {
        "name": "roster position",
        "rawType": "object",
        "type": "string"
```

```
 },
 {
  "name": "salary",
  "rawType": "int64",
  "type": "integer"
 },
 {
  "name": "game info",
  "rawType": "object",
  "type": "string"
 },
 {
  "name": "team",
  "rawType": "object",
  "type": "string"
 },
 {
  "name": "avgpointspergame",
  "rawType": "float64",
  "type": "float"
 },
 {
  "name": "projpts",
  "rawType": "float64",
  "type": "float"
 },
 {
  "name": "projown",
  "rawType": "float64",
  "type": "float"
 }
],
"ref": "23a0ab4f-42b7-4a2c-85d6-088a35572c0b",
"rows": [
 [
  "0",
  "40142470",
  "Christian McCaffrey",
  "RB",
  "RB/FLEX",
  "8500",
  "JAX@SF 09/28/2025 04:05PM ET",
  "49ers",
  "23.3",
  "3.373679906475014",
  "7.360534456183173"
 ],
 [
  "1",
  "40142472",
  "Bijan Robinson",
  "RB",
```

```
   "RB/FLEX",
   "8200",
   "WAS@ATL 09/28/2025 01:00PM ET",
   "Falcons",
   "22.1",
   "3.1018619303914563",
   "3.4307166971845753"
  ],
  [
   "2",
   "40142716",
   "Puka Nacua",
   "WR",
   "WR/FLEX",
   "7900",
   "IND@LAR 09/28/2025 04:05PM ET",
   "Rams",
   "26.5",
   "2.7394379622800464",
   "8.913055052330767"
  ]
 ],
 "shape": {
  "columns": 10,
  "rows": 3
 }
},
"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>id</th>\n",
 "      <th>name</th>\n",
 "      <th>position</th>\n",
 "      <th>roster position</th>\n",
 "      <th>salary</th>\n",
 "      <th>game info</th>\n",
```

```
 "      <th>team</th>\n",
 "      <th>avgpointspergame</th>\n",
 "      <th>projpts</th>\n",
 "      <th>projown</th>\n",
 "    </tr>\n",
 "  </thead>\n",
 "  <tbody>\n",
 "    <tr>\n",
 "      <th>0</th>\n",
 "      <td>40142470</td>\n",
 "      <td>Christian McCaffrey</td>\n",
 "      <td>RB</td>\n",
 "      <td>RB/FLEX</td>\n",
 "      <td>8500</td>\n",
 "      <td>JAX@SF 09/28/2025 04:05PM ET</td>\n",
 "      <td>49ers</td>\n",
 "      <td>23.3</td>\n",
 "      <td>3.373680</td>\n",
 "      <td>7.360534</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>1</th>\n",
 "      <td>40142472</td>\n",
 "      <td>Bijan Robinson</td>\n",
 "      <td>RB</td>\n",
 "      <td>RB/FLEX</td>\n",
 "      <td>8200</td>\n",
 "      <td>WAS@ATL 09/28/2025 01:00PM ET</td>\n",
 "      <td>Falcons</td>\n",
 "      <td>22.1</td>\n",
 "      <td>3.101862</td>\n",
 "      <td>3.430717</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>2</th>\n",
 "      <td>40142716</td>\n",
 "      <td>Puka Nacua</td>\n",
 "      <td>WR</td>\n",
 "      <td>WR/FLEX</td>\n",
 "      <td>7900</td>\n",
 "      <td>IND@LAR 09/28/2025 04:05PM ET</td>\n",
 "      <td>Rams</td>\n",
 "      <td>26.5</td>\n",
 "      <td>2.739438</td>\n",
 "      <td>8.913055</td>\n",
 "    </tr>\n",
 "  </tbody>\n",
 "</table>\n",
 "</div>"
],
"text/plain": [
 "         id                name position roster position  salary  \\\n",
```

```
       "0  40142470  Christian McCaffrey       RB         RB/FLEX    8500   \n",
       "1  40142472       Bijan Robinson       RB         RB/FLEX    8200   \n",
       "2  40142716           Puka Nacua       WR         WR/FLEX    7900   \n",
       "\n",
       "                        game info     team  avgpointspergame   projpts  \\\n",
       "0   JAX@SF 09/28/2025 04:05PM ET    49ers              23.3  3.373680   \n",
       "1  WAS@ATL 09/28/2025 01:00PM ET  Falcons              22.1  3.101862   \n",
       "2  IND@LAR 09/28/2025 04:05PM ET     Rams              26.5  2.739438   \n",
       "\n",
       "    projown  \n",
       "0  7.360534  \n",
       "1  3.430717  \n",
       "2  8.913055  "
      ]
     },
     "execution_count": 3,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "player_data = normalize_features(player_data, ['projown', 'projpts'])\n",
    "player_data.head(3)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "id": "862f06c0",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Lineup #1: Score=39.73, Salary=49900, projpts=16.572887171635976, projown=42.30203153693488,
Locked=OK.\n",    id                 name position roster position  salary  \\\n",
      "0    40142470  Christian McCaffrey       RB         RB/FLEX    8500   \n",
      "2    40142716           Puka Nacua       WR         WR/FLEX    7900   \n",
      "37   40142494      Omarion Hampton       RB         RB/FLEX    5900   \n",
      "53   40142504         Cam Skattebo       RB         RB/FLEX    5500   \n",
      "58   40142762        Jakobi Meyers       WR         WR/FLEX    5400   \n",
      "67   40142772          Chris Olave       WR         WR/FLEX    5100   \n",
      "132  40142808           Tre Tucker       WR         WR/FLEX    4200   \n",
      "193  40143150         Hunter Henry       TE         TE/FLEX    4000   \n",
      "286  40143368              Texans       DST             DST    3400   \n",
      "\n",
      "                        game info      team  avgpointspergame   projpts  \\\n",
      "0     JAX@SF 09/28/2025 04:05PM ET     49ers             23.30  3.373680   \n",
      "2    IND@LAR 09/28/2025 04:05PM ET      Rams             26.50  2.739438   \n",
      "37   LAC@NYG 09/28/2025 01:00PM ET  Chargers             12.17  1.996469   \n",
      "53   LAC@NYG 09/28/2025 01:00PM ET    Giants             13.63  1.797136   \n",
```

```
      "58    CHI@LV 09/28/2025 04:25PM ET   Raiders             13.27  1.942105   \n",
      "67    NO@BUF 09/28/2025 01:00PM ET    Saints             13.17  1.688408   \n",
      "132   CHI@LV 09/28/2025 04:25PM ET   Raiders             20.73  1.036045   \n",
      "193   CAR@NE 09/28/2025 01:00PM ET  Patriots             13.83  1.307863   \n",
      "286  TEN@HOU 09/28/2025 01:00PM ET    Texans              6.00  0.691743   \n",
      "\n",
      "      projown  \n",
      "0    7.360534  \n",
      "2    8.913055  \n",
      "37   4.449558  \n",
      "53   3.018328  \n",
      "58   5.274335  \n",
      "67   4.085686  \n",
      "132  3.139619  \n",
      "193  3.066845  \n",
      "286  2.994070  \n",
      "----------------------------------\n",
      "Lineup #2: Score=39.64, Salary=49300, projpts=16.518523576419266, projown=42.20499899967565,
Locked OK.\n",    id                 name position roster position  salary  \\\n",
      "0    40142470  Christian McCaffrey       RB         RB/FLEX    8500   \n",
      "2    40142716           Puka Nacua       WR         WR/FLEX    7900   \n",
      "11   40142482       James Cook III       RB         RB/FLEX    7100   \n",
      "37   40142494      Omarion Hampton       RB         RB/FLEX    5900   \n",
      "58   40142762        Jakobi Meyers       WR         WR/FLEX    5400   \n",
      "67   40142772          Chris Olave       WR         WR/FLEX    5100   \n",
      "193  40143150         Hunter Henry       TE         TE/FLEX    4000   \n",
      "308  40142878      Adonai Mitchell       WR         WR/FLEX    3100   \n",
      "540  40143385              Titans       DST             DST    2300   \n",
      "\n",
      "                        game info      team  avgpointspergame   projpts  \\\n",
      "0     JAX@SF 09/28/2025 04:05PM ET     49ers             23.30  3.373680   \n",
      "2    IND@LAR 09/28/2025 04:05PM ET      Rams             26.50  2.739438   \n",
      "11    NO@BUF 09/28/2025 01:00PM ET     Bills             24.83  2.540105   \n",
      "37   LAC@NYG 09/28/2025 01:00PM ET  Chargers             12.17  1.996469   \n",
      "58    CHI@LV 09/28/2025 04:25PM ET   Raiders             13.27  1.942105   \n",
      "67    NO@BUF 09/28/2025 01:00PM ET    Saints             13.17  1.688408   \n",
      "193   CAR@NE 09/28/2025 01:00PM ET  Patriots             13.83  1.307863   \n",
      "308  IND@LAR 09/28/2025 04:05PM ET     Colts              2.70  0.655500   \n",
      "540  TEN@HOU 09/28/2025 01:00PM ET    Titans              3.33  0.274955   \n",
      "\n",
      "      projown  \n",
      "0    7.360534  \n",
      "2    8.913055  \n",
      "11   4.692140  \n",
      "37   4.449558  \n",
      "58   5.274335  \n",
      "67   4.085686  \n",
      "193  3.066845  \n",
      "308  1.756905  \n",
      "540  2.605940  \n",
      "----------------------------------\n",
      "Lineup #3: Score=39.61, Salary=50000, projpts=16.69973556047497, projown=42.15648273104604,
Locked OK.\n",
```

NB_dkCashLineup.ipynb (page 11)

```
"           id                 name position roster position  salary  \\\n",
"0    40142470  Christian McCaffrey       RB         RB/FLEX    8500   \n",
"2    40142716           Puka Nacua       WR         WR/FLEX    7900   \n",
"11   40142482       James Cook III       RB         RB/FLEX    7100   \n",
"53   40142504         Cam Skattebo       RB         RB/FLEX    5500   \n",
"58   40142762        Jakobi Meyers       WR         WR/FLEX    5400   \n",
"67   40142772          Chris Olave       WR         WR/FLEX    5100   \n",
"132  40142808           Tre Tucker       WR         WR/FLEX    4200   \n",
"193  40143150         Hunter Henry       TE         TE/FLEX    4000   \n",
"540  40143385              Titans       DST             DST    2300   \n",
"\n",
"                         game info      team  avgpointspergame   projpts  \\\n",
"0     JAX@SF 09/28/2025 04:05PM ET     49ers             23.30  3.373680   \n",
"2    IND@LAR 09/28/2025 04:05PM ET      Rams             26.50  2.739438   \n",
"11    NO@BUF 09/28/2025 01:00PM ET     Bills             24.83  2.540105   \n",
"53   LAC@NYG 09/28/2025 01:00PM ET    Giants             13.63  1.797136   \n",
"58    CHI@LV 09/28/2025 04:25PM ET   Raiders             13.27  1.942105   \n",
"67    NO@BUF 09/28/2025 01:00PM ET    Saints             13.17  1.688408   \n",
"132   CHI@LV 09/28/2025 04:25PM ET   Raiders             20.73  1.036045   \n",
"193   CAR@NE 09/28/2025 01:00PM ET  Patriots             13.83  1.307863   \n",
"540  TEN@HOU 09/28/2025 01:00PM ET    Titans              3.33  0.274955   \n",
"\n",
"      projown  \n",
"0    7.360534  \n",
"2    8.913055  \n",
"11   4.692140  \n",
"53   3.018328  \n",
"58   5.274335  \n",
"67   4.085686  \n",
"132  3.139619  \n",
"193  3.066845  \n",
"540  2.605940  \n",
"----------------------------------\n",
"Lineup #4: Score=39.59, Salary=49600, projpts=16.083614814685575, projown=42.20499899967565,
Locked OK.\n",    id                 name position roster position  salary  \\\n",
"0    40142470  Christian McCaffrey       RB         RB/FLEX    8500   \n",
"2    40142716           Puka Nacua       WR         WR/FLEX    7900   \n",
"24   40142486       Kyren Williams       RB         RB/FLEX    6300   \n",
"37   40142494      Omarion Hampton       RB         RB/FLEX    5900   \n",
"58   40142762        Jakobi Meyers       WR         WR/FLEX    5400   \n",
"67   40142772          Chris Olave       WR         WR/FLEX    5100   \n",
"132  40142808           Tre Tucker       WR         WR/FLEX    4200   \n",
"193  40143150         Hunter Henry       TE         TE/FLEX    4000   \n",
"540  40143385              Titans       DST             DST    2300   \n",
"\n",
"                         game info      team  avgpointspergame   projpts  \\\n",
"0     JAX@SF 09/28/2025 04:05PM ET     49ers             23.30  3.373680   \n",
"2    IND@LAR 09/28/2025 04:05PM ET      Rams             26.50  2.739438   \n",
"24   IND@LAR 09/28/2025 04:05PM ET      Rams             14.37  1.724651   \n",
"37   LAC@NYG 09/28/2025 01:00PM ET  Chargers             12.17  1.996469   \n",
"58    CHI@LV 09/28/2025 04:25PM ET   Raiders             13.27  1.942105   \n",
"67    NO@BUF 09/28/2025 01:00PM ET    Saints             13.17  1.688408   \n",
```

NB_dkCashLineup.ipynb (page 12)

```
      "132   CHI@LV 09/28/2025 04:25PM ET   Raiders             20.73  1.036045   \n",
      "193   CAR@NE 09/28/2025 01:00PM ET  Patriots             13.83  1.307863   \n",
      "540  TEN@HOU 09/28/2025 01:00PM ET    Titans              3.33  0.274955   \n",
      "\n",
      "      projown  \n",
      "0    7.360534  \n",
      "2    8.913055  \n",
      "24   3.309426  \n",
      "37   4.449558  \n",
      "58   5.274335  \n",
      "67   4.085686  \n",
      "132  3.139619  \n",
      "193  3.066845  \n",
      "540  2.605940  \n",
      "-----------------------------------\n",
      "Lineup #5: Score=39.56, Salary=49900, projpts=16.446038782796986, projown=42.132224596731234,
Locked OK.\n",    id                 name position roster position  salary  \\\n",
      "0    40142470  Christian McCaffrey       RB         RB/FLEX    8500   \n",
      "2    40142716           Puka Nacua       WR         WR/FLEX    7900   \n",
      "11   40142482       James Cook III       RB         RB/FLEX    7100   \n",
      "37   40142494      Omarion Hampton       RB         RB/FLEX    5900   \n",
      "58   40142762        Jakobi Meyers       WR         WR/FLEX    5400   \n",
      "67   40142772          Chris Olave       WR         WR/FLEX    5100   \n",
      "98   40143138         Tyler Warren       TE         TE/FLEX    4600   \n",
      "308  40142878      Adonai Mitchell       WR         WR/FLEX    3100   \n",
      "540  40143385               Titans      DST             DST    2300   \n",
      "\n",
      "                         game info      team  avgpointspergame   projpts  \\\n",
      "0     JAX@SF 09/28/2025 04:05PM ET     49ers             23.30  3.373680   \n",
      "2    IND@LAR 09/28/2025 04:05PM ET      Rams             26.50  2.739438   \n",
      "11    NO@BUF 09/28/2025 01:00PM ET     Bills             24.83  2.540105   \n",
      "37   LAC@NYG 09/28/2025 01:00PM ET  Chargers             12.17  1.996469   \n",
      "58    CHI@LV 09/28/2025 04:25PM ET   Raiders             13.27  1.942105   \n",
      "67    NO@BUF 09/28/2025 01:00PM ET    Saints             13.17  1.688408   \n",
      "98   IND@LAR 09/28/2025 04:05PM ET     Colts             11.20  1.235378   \n",
      "308  IND@LAR 09/28/2025 04:05PM ET     Colts              2.70  0.655500   \n",
      "540  TEN@HOU 09/28/2025 01:00PM ET    Titans              3.33  0.274955   \n",
      "\n",
      "      projown  \n",
      "0    7.360534  \n",
      "2    8.913055  \n",
      "11   4.692140  \n",
      "37   4.449558  \n",
      "58   5.274335  \n",
      "67   4.085686  \n",
      "98   2.994070  \n",
      "308  1.756905  \n",
      "540  2.605940  \n",
      "-----------------------------------\n"
    ]
   }
  ],
```

NB_dkCashLineup.ipynb (page 13)

```
   "source": [
    "lineups = generate_top_lineups(player_data, \n",
    "                               num_lineups=5,\n",
    "                               locked_players=([40142470, 40142716]),\n",
    "                               projown_wt=0.1,\n",
    "                               projpts_wt=0.9\n",
    "                              )\n",
    "\n",
    "for i, lineup in enumerate(lineups):\n",
    "    print(f\"Lineup #{i+1}: Score={lineup['score']:.2f},
Salary={lineup['lineup']['salary'].sum()}, projpts={lineup['lineup']['projpts'].sum()},
projown={lineup['lineup']['projown'].sum()}, Locked OK.\")\n",
    "    print(\"-\" * 35)"
   ]
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "FantasySports_py3_12_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": "3.12.3"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```

NB_dkCashReview.ipynb (page 1)


```
{
 "cells": [
  {
   "cell_type": "markdown",
   "id": "f83f6f86",
   "metadata": {},
   "source": [
    "**Objective**: Analyze DraftKings DFS contest data to predict successful lineups for cash games (top 50%).\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "027527f1",
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd\n",
    "import numpy as np\n",
    "from utils.draftkingsCashReview import load_split_dk_results, load_player_data, add_lineup_features, calculate_lineup_rating, normalize_features\n",
    "\n"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "a81f3a83",
   "metadata": {},
   "source": [
    "**Input Data**:\n",
    "1. `contest-standings.csv` load csv and split at column 6 into 2 df\n",
    "2. `entries_df` (contest results):  \n",
    "   Columns: `Rank`, `EntryId`, `EntryName`, `TimeRemaining`, `Points`, `Lineup`, `cash_win`, `cash_result`\n",
    "3. `player_results_df` (player projections):  \n",
    "   Columns: `Player`, `RosterPosition`, `%Drafted`, `FPTS`, `teamabbrev`, `avgpointspergame`, `projpts`, `projown`, `salary`.\n",
    "\n",
    "**Steps to Execute**:\n",
    "1. **[Data Prep]**:  \n",
    "   - Clean `Player` strings in both DataFrames.  \n",
    "   - Parse `Lineup` strings into individual player lists.  \n",
    "   - Compute new features per lineup (stacks, value exposure, etc.).\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "d87f8414",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "application/vnd.microsoft.datawrangler.viewer.v0+json": {
       "columns": [
```

```
{
 "name": "index",
 "rawType": "int64",
 "type": "integer"
},
{
 "name": "Player",
 "rawType": "object",
 "type": "string"
},
{
 "name": "Roster Position",
 "rawType": "object",
 "type": "string"
},
{
 "name": "%Drafted",
 "rawType": "object",
 "type": "string"
},
{
 "name": "FPTS",
 "rawType": "float64",
 "type": "float"
},
{
 "name": "position",
 "rawType": "object",
 "type": "string"
},
{
 "name": "name + id",
 "rawType": "object",
 "type": "string"
},
{
 "name": "name",
 "rawType": "object",
 "type": "string"
},
{
 "name": "id",
 "rawType": "float64",
 "type": "float"
},
{
 "name": "roster position",
 "rawType": "object",
 "type": "string"
},
{
 "name": "salary",
```

NB_dkCashReview.ipynb (page 3)


```
  "rawType": "float64",
  "type": "float"
 },
 {
  "name": "game info",
  "rawType": "object",
  "type": "string"
 },
 {
  "name": "teamabbrev",
  "rawType": "object",
  "type": "string"
 },
 {
  "name": "avgpointspergame",
  "rawType": "float64",
  "type": "float"
 },
 {
  "name": "team",
  "rawType": "object",
  "type": "string"
 },
 {
  "name": "projpts",
  "rawType": "float64",
  "type": "float"
 },
 {
  "name": "projown",
  "rawType": "float64",
  "type": "float"
 }
],
"ref": "49dfa012-faa7-4c71-bfa3-51c7159a5945",
"rows": [
 [
  "0",
  "Christian McCaffrey",
  "RB",
  "72.56%",
  "22.7",
  "RB",
  "Christian McCaffrey (39971377)",
  "Christian McCaffrey",
  "39971377.0",
  "RB/FLEX",
  "7500.0",
  "SF@NO 09/14/2025 01:00PM ET",
  "SF",
  "23.2",
  "49ers",
```

```
   "19.5",
   "34.7"
  ],
  [
   "1",
   "Hollywood Brown",
   "WR",
   "52.76%",
   "8.0",
   "WR",
   "Hollywood Brown (39971707)",
   "Hollywood Brown",
   "39971707.0",
   "WR/FLEX",
   "5200.0",
   "PHI@KC 09/14/2025 04:25PM ET",
   "KC",
   "19.9",
   "Chiefs",
   "14.0",
   "15.7"
  ],
  [
   "2",
   "Harold Fannin Jr.",
   "TE",
   "50.17%",
   "9.8",
   "TE",
   "Harold Fannin Jr. (39972149)",
   "Harold Fannin Jr.",
   "39972149.0",
   "TE/FLEX",
   "3100.0",
   "CLE@BAL 09/14/2025 01:00PM ET",
   "CLE",
   "13.6",
   "Browns",
   "9.6",
   "13.3"
  ]
 ],
 "shape": {
  "columns": 16,
  "rows": 3
 }
},
"text/html": [
 "<div>\n",
 "<style scoped>\n",
 "    .dataframe tbody tr th:only-of-type {\n",
 "        vertical-align: middle;\n",
```

NB_dkCashReview.ipynb (page 5)

```
"    }\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>Player</th>\n",
"      <th>Roster Position</th>\n",
"      <th>%Drafted</th>\n",
"      <th>FPTS</th>\n",
"      <th>position</th>\n",
"      <th>name + id</th>\n",
"      <th>name</th>\n",
"      <th>id</th>\n",
"      <th>roster position</th>\n",
"      <th>salary</th>\n",
"      <th>game info</th>\n",
"      <th>teamabbrev</th>\n",
"      <th>avgpointspergame</th>\n",
"      <th>team</th>\n",
"      <th>projpts</th>\n",
"      <th>projown</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>Christian McCaffrey</td>\n",
"      <td>RB</td>\n",
"      <td>72.56%</td>\n",
"      <td>22.7</td>\n",
"      <td>RB</td>\n",
"      <td>Christian McCaffrey (39971377)</td>\n",
"      <td>Christian McCaffrey</td>\n",
"      <td>39971377.0</td>\n",
"      <td>RB/FLEX</td>\n",
"      <td>7500.0</td>\n",
"      <td>SF@NO 09/14/2025 01:00PM ET</td>\n",
"      <td>SF</td>\n",
"      <td>23.2</td>\n",
"      <td>49ers</td>\n",
"      <td>19.5</td>\n",
"      <td>34.7</td>\n",
"    </tr>\n",
```

NB_dkCashReview.ipynb (page 6)


```
 "    <tr>\n",
 "      <th>1</th>\n",
 "      <td>Hollywood Brown</td>\n",
 "      <td>WR</td>\n",
 "      <td>52.76%</td>\n",
 "      <td>8.0</td>\n",
 "      <td>WR</td>\n",
 "      <td>Hollywood Brown (39971707)</td>\n",
 "      <td>Hollywood Brown</td>\n",
 "      <td>39971707.0</td>\n",
 "      <td>WR/FLEX</td>\n",
 "      <td>5200.0</td>\n",
 "      <td>PHI@KC 09/14/2025 04:25PM ET</td>\n",
 "      <td>KC</td>\n",
 "      <td>19.9</td>\n",
 "      <td>Chiefs</td>\n",
 "      <td>14.0</td>\n",
 "      <td>15.7</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>2</th>\n",
 "      <td>Harold Fannin Jr.</td>\n",
 "      <td>TE</td>\n",
 "      <td>50.17%</td>\n",
 "      <td>9.8</td>\n",
 "      <td>TE</td>\n",
 "      <td>Harold Fannin Jr. (39972149)</td>\n",
 "      <td>Harold Fannin Jr.</td>\n",
 "      <td>39972149.0</td>\n",
 "      <td>TE/FLEX</td>\n",
 "      <td>3100.0</td>\n",
 "      <td>CLE@BAL 09/14/2025 01:00PM ET</td>\n",
 "      <td>CLE</td>\n",
 "      <td>13.6</td>\n",
 "      <td>Browns</td>\n",
 "      <td>9.6</td>\n",
 "      <td>13.3</td>\n",
 "    </tr>\n",
 "  </tbody>\n",
 "</table>\n",
 "</div>"
],
"text/plain": [
 "                Player Roster Position %Drafted  FPTS position  \\\n",
 "0  Christian McCaffrey              RB   72.56%  22.7       RB   \n",
 "1      Hollywood Brown              WR   52.76%   8.0       WR   \n",
 "2    Harold Fannin Jr.              TE   50.17%   9.8       TE   \n",
 "\n",
 "                        name + id                 name          id  \\\n",
 "0  Christian McCaffrey (39971377)  Christian McCaffrey  39971377.0   \n",
 "1      Hollywood Brown (39971707)      Hollywood Brown  39971707.0   \n",
 "2    Harold Fannin Jr. (39972149)    Harold Fannin Jr.  39972149.0   \n",
```

NB_dkCashReview.ipynb (page 7)


```
     "\n",
     "  roster position  salary                      game info teamabbrev  \\\n",
     "0         RB/FLEX  7500.0    SF@NO 09/14/2025 01:00PM ET         SF   \n",
     "1         WR/FLEX  5200.0   PHI@KC 09/14/2025 04:25PM ET         KC   \n",
     "2         TE/FLEX  3100.0  CLE@BAL 09/14/2025 01:00PM ET        CLE   \n",
     "\n",
     "   avgpointspergame    team  projpts  projown  \n",
     "0              23.2   49ers     19.5     34.7  \n",
     "1              19.9  Chiefs     14.0     15.7  \n",
     "2              13.6  Browns      9.6     13.3  "
    ]
   },
   "execution_count": 2,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "# Load the data\n",
  "week = 'week02'\n",
  "results_file_name = 'contest-standings.csv'\n",
  "playerpool_df, entries_df = load_split_dk_results(week)\n",
  "player_results_df = load_player_data(week, playerpool_df)\n",
  "\n",
  "player_results_df.head(3)  "
 ]
},
{
 "cell_type": "code",
 "execution_count": 4,
 "id": "dfc6f760",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "application/vnd.microsoft.datawrangler.viewer.v0+json": {
     "columns": [
      {
       "name": "index",
       "rawType": "int64",
       "type": "integer"
      },
      {
       "name": "Rank",
       "rawType": "int64",
       "type": "integer"
      },
      {
       "name": "EntryId",
       "rawType": "int64",
       "type": "integer"
      },
```

NB_dkCashReview.ipynb (page 8)


```
        {
         "name": "EntryName",
         "rawType": "object",
         "type": "string"
        },
        {
         "name": "TimeRemaining",
         "rawType": "int64",
         "type": "integer"
        },
        {
         "name": "Points",
         "rawType": "float64",
         "type": "float"
        },
        {
         "name": "Lineup",
         "rawType": "object",
         "type": "string"
        }
       ],
       "ref": "15469993-dea5-4513-a041-29a75980b063",
       "rows": [
        [
         "0",
         "1",
         "4852565135",
         "calebzsnyder",
         "0",
         "206.86002",
         "DST Ravens  FLEX George Pickens QB Jared Goff RB Christian McCaffrey RB Breece Hall TE
Harold Fannin Jr. WR Amon-Ra St. Brown WR Jameson Williams WR Rome Odunze"
        ],
        [
         "1",
         "2",
         "4852356677",
         "trh1010",
         "0",
         "204.36002",
         "DST Colts  FLEX Ja'Marr Chase QB Mac Jones RB Christian McCaffrey RB Javonte Williams TE
Harold Fannin Jr. WR CeeDee Lamb WR Davante Adams WR Rome Odunze"
        ],
        [
         "2",
         "3",
         "4853930626",
         "mbhurls",
         "0",
         "200.76",
         "DST Cowboys  FLEX De'Von Achane QB Jared Goff RB Jahmyr Gibbs RB James Cook TE Harold
Fannin Jr. WR Amon-Ra St. Brown WR Jaxon Smith-Njigba WR DeAndre Hopkins"
        ],
       ],
       "shape": {
```

```
  "columns": 6,
  "rows": 3
 }
},
"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>Rank</th>\n",
 "      <th>EntryId</th>\n",
 "      <th>EntryName</th>\n",
 "      <th>TimeRemaining</th>\n",
 "      <th>Points</th>\n",
 "      <th>Lineup</th>\n",
 "    </tr>\n",
 "  </thead>\n",
 "  <tbody>\n",
 "    <tr>\n",
 "      <th>0</th>\n",
 "      <td>1</td>\n",
 "      <td>4852565135</td>\n",
 "      <td>calebzsnyder</td>\n",
 "      <td>0</td>\n",
 "      <td>206.86002</td>\n",
 "      <td>DST Ravens  FLEX George Pickens QB Jared Goff ...</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>1</th>\n",
 "      <td>2</td>\n",
 "      <td>4852356677</td>\n",
 "      <td>trh1010</td>\n",
 "      <td>0</td>\n",
 "      <td>204.36002</td>\n",
 "      <td>DST Colts  FLEX Ja'Marr Chase QB Mac Jones RB ...</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>2</th>\n",
```

```
     "      <td>3</td>\n",
     "      <td>4853930626</td>\n",
     "      <td>mbhurls</td>\n",
     "      <td>0</td>\n",
     "      <td>200.76000</td>\n",
     "      <td>DST Cowboys  FLEX De'Von Achane QB Jared Goff ...</td>\n",
     "    </tr>\n",
     "  </tbody>\n",
     "</table>\n",
     "</div>"
    ],
    "text/plain": [
     "   Rank     EntryId     EntryName  TimeRemaining     Points  \\\n",
     "0     1  4852565135  calebzsnyder              0  206.86002   \n",
     "1     2  4852356677       trh1010              0  204.36002   \n",
     "2     3  4853930626       mbhurls              0  200.76000   \n",
     "\n",
     "                                            Lineup  \n",
     "0  DST Ravens  FLEX George Pickens QB Jared Goff ...  \n",
     "1  DST Colts  FLEX Ja'Marr Chase QB Mac Jones RB ...  \n",
     "2  DST Cowboys  FLEX De'Von Achane QB Jared Goff ...  "
    ]
   },
   "execution_count": 4,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "entries_df.head(3)"
 ]
},
{
 "cell_type": "markdown",
 "id": "cbfb579a",
 "metadata": {},
 "source": [
  "\n",
  "2. **[Feature Engineering]**:  \n",
  "   - Expand lineup\n",
  "   - For each lineup, create features like:  \n",
  "     - `sum_projpts`, `sum_projown`  \n",
  "     - `num_stacks`, `num_value_players` (projown < 20%)  \n",
  "     - `avg_floor` (using avgpointspergame)  \n",
  "     - `qb_wr_stacks`  \n",
  "     - `num_chalk_players` (projown > 60%)  \n",
  "   - Export to `training_data.csv`.\n"
 ]
},
{
 "cell_type": "code",
 "execution_count": 5,
```

NB_dkCashReview.ipynb (page 11)


```
   "id": "a935f204",
   "metadata": {},
   "outputs": [
    {
     "ename": "TypeError",
     "evalue": "'float' object is not iterable",
     "output_type": "error",
     "traceback": [
"\u001b[1;31m---------------------------------------------------------------------------\u001b[0m",
      "\u001b[1;31mTypeError\u001b[0m                                 Traceback (most recent call last)",
"Cell \u001b[1;32mIn[5], line 1\u001b[0m\n\u001b[1;32m----> 1\u001b[0m player_results_df
```

```
},
{
 "name": "%Drafted",
 "rawType": "object",
 "type": "string"
},
{
 "name": "FPTS",
 "rawType": "float64",
 "type": "float"
},
{
 "name": "position",
 "rawType": "object",
 "type": "string"
},
{
 "name": "name + id",
 "rawType": "object",
 "type": "string"
},
{
 "name": "name",
 "rawType": "object",
 "type": "string"
},
{
 "name": "id",
 "rawType": "float64",
 "type": "float"
},
{
 "name": "roster position",
 "rawType": "object",
 "type": "string"
},
{
 "name": "salary",
 "rawType": "float64",
 "type": "float"
},
{
 "name": "game info",
 "rawType": "object",
 "type": "string"
},
{
 "name": "teamabbrev",
 "rawType": "object",
 "type": "string"
},
{
```

```
  "name": "avgpointspergame",
  "rawType": "float64",
  "type": "float"
 },
 {
  "name": "team",
  "rawType": "object",
  "type": "string"
 },
 {
  "name": "projpts",
  "rawType": "float64",
  "type": "float"
 },
 {
  "name": "projown",
  "rawType": "float64",
  "type": "float"
 }
],
"ref": "a4660628-a419-410d-b675-3dc0ad076709",
"rows": [
 [
  "0",
  "Christian McCaffrey",
  "RB",
  "72.56%",
  "22.7",
  "RB",
  "Christian McCaffrey (39971377)",
  "Christian McCaffrey",
  "39971377.0",
  "RB/FLEX",
  "7500.0",
  "SF@NO 09/14/2025 01:00PM ET",
  "SF",
  "23.2",
  "49ers",
  "1.8819230649982939",
  "6.232744368208733"
 ],
 [
  "1",
  "Hollywood Brown",
  "WR",
  "52.76%",
  "8.0",
  "WR",
  "Hollywood Brown (39971707)",
  "Hollywood Brown",
  "39971707.0",
  "WR/FLEX",
```

```
   "5200.0",
   "PHI@KC 09/14/2025 04:25PM ET",
   "KC",
   "19.9",
   "Chiefs",
   "0.8193367723424522",
   "2.4030318908881454"
  ],
  [
   "2",
   "Harold Fannin Jr.",
   "TE",
   "50.17%",
   "9.8",
   "TE",
   "Harold Fannin Jr. (39972149)",
   "Harold Fannin Jr.",
   "39972149.0",
   "TE/FLEX",
   "3100.0",
   "CLE@BAL 09/14/2025 01:00PM ET",
   "CLE",
   "13.6",
   "Browns",
   "-0.030732261782221344",
   "1.9192787358581767"
  ]
 ],
 "shape": {
  "columns": 16,
  "rows": 3
 }
},
"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",
```

NB_dkCashReview.ipynb (page 15)

```
"      <th>Player</th>\n",
"      <th>Roster Position</th>\n",
"      <th>%Drafted</th>\n",
"      <th>FPTS</th>\n",
"      <th>position</th>\n",
"      <th>name + id</th>\n",
"      <th>name</th>\n",
"      <th>id</th>\n",
"      <th>roster position</th>\n",
"      <th>salary</th>\n",
"      <th>game info</th>\n",
"      <th>teamabbrev</th>\n",
"      <th>avgpointspergame</th>\n",
"      <th>team</th>\n",
"      <th>projpts</th>\n",
"      <th>projown</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>Christian McCaffrey</td>\n",
"      <td>RB</td>\n",
"      <td>72.56%</td>\n",
"      <td>22.7</td>\n",
"      <td>RB</td>\n",
"      <td>Christian McCaffrey (39971377)</td>\n",
"      <td>Christian McCaffrey</td>\n",
"      <td>39971377.0</td>\n",
"      <td>RB/FLEX</td>\n",
"      <td>7500.0</td>\n",
"      <td>SF@NO 09/14/2025 01:00PM ET</td>\n",
"      <td>SF</td>\n",
"      <td>23.2</td>\n",
"      <td>49ers</td>\n",
"      <td>1.881923</td>\n",
"      <td>6.232744</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>Hollywood Brown</td>\n",
"      <td>WR</td>\n",
"      <td>52.76%</td>\n",
"      <td>8.0</td>\n",
"      <td>WR</td>\n",
"      <td>Hollywood Brown (39971707)</td>\n",
"      <td>Hollywood Brown</td>\n",
"      <td>39971707.0</td>\n",
"      <td>WR/FLEX</td>\n",
"      <td>5200.0</td>\n",
"      <td>PHI@KC 09/14/2025 04:25PM ET</td>\n",
"      <td>KC</td>\n",
```

```
  "      <td>19.9</td>\n",
  "      <td>Chiefs</td>\n",
  "      <td>0.819337</td>\n",
  "      <td>2.403032</td>\n",
  "    </tr>\n",
  "    <tr>\n",
  "      <th>2</th>\n",
  "      <td>Harold Fannin Jr.</td>\n",
  "      <td>TE</td>\n",
  "      <td>50.17%</td>\n",
  "      <td>9.8</td>\n",
  "      <td>TE</td>\n",
  "      <td>Harold Fannin Jr. (39972149)</td>\n",
  "      <td>Harold Fannin Jr.</td>\n",
  "      <td>39972149.0</td>\n",
  "      <td>TE/FLEX</td>\n",
  "      <td>3100.0</td>\n",
  "      <td>CLE@BAL 09/14/2025 01:00PM ET</td>\n",
  "      <td>CLE</td>\n",
  "      <td>13.6</td>\n",
  "      <td>Browns</td>\n",
  "      <td>-0.030732</td>\n",
  "      <td>1.919279</td>\n",
  "    </tr>\n",
  "  </tbody>\n",
  "</table>\n",
  "</div>"
 ],
 "text/plain": [
  "                Player Roster Position %Drafted  FPTS position  \\\n",
  "0  Christian McCaffrey              RB   72.56%  22.7       RB   \n",
  "1      Hollywood Brown              WR   52.76%   8.0       WR   \n",
  "2    Harold Fannin Jr.              TE   50.17%   9.8       TE   \n",
  "\n",
  "                         name + id                 name          id  \\\n",
  "0  Christian McCaffrey (39971377)  Christian McCaffrey  39971377.0   \n",
  "1      Hollywood Brown (39971707)      Hollywood Brown  39971707.0   \n",
  "2    Harold Fannin Jr. (39972149)    Harold Fannin Jr.  39972149.0   \n",
  "\n",
  "  roster position  salary                      game info teamabbrev  \\\n",
  "0         RB/FLEX  7500.0    SF@NO 09/14/2025 01:00PM ET         SF   \n",
  "1         WR/FLEX  5200.0   PHI@KC 09/14/2025 04:25PM ET         KC   \n",
  "2         TE/FLEX  3100.0  CLE@BAL 09/14/2025 01:00PM ET        CLE   \n",
  "\n",
  "   avgpointspergame    team   projpts   projown  \n",
  "0              23.2   49ers  1.881923  6.232744  \n",
  "1              19.9  Chiefs  0.819337  2.403032  \n",
  "2              13.6  Browns -0.030732  1.919279  "
 ]
},
"execution_count": 14,
"metadata": {},
```

NB_dkCashReview.ipynb (page 17)


```
   "output_type": "execute_result"
  }
 ],
 "source": [
  "player_results_df = normalize_features(player_results_df, ['projown', 'projpts'])\n",
  "player_results_df.head(3)"
 ]
},
{
 "cell_type": "markdown",
 "id": "bc12a15e",
 "metadata": {},
 "source": [
  "\n",
  "3. **[Model Training]**:  \n",
  "   - **Regression**:  \n",
  "     Train `LinearRegression` to predict `Points` from lineup features.  \n",
  "     - Save coefficients as `reg_weights.csv`.  \n",
  "   - **Classification**:  \n",
  "     Train `LogisticRegression` to predict `cash_win` from lineup features.  \n",
  "     - Save coefficients as `logit_weights.csv`.  \n"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "id": "adcc2271",
 "metadata": {},
 "outputs": [],
 "source": []
},
{
 "cell_type": "code",
 "execution_count": null,
 "id": "0619a56b",
 "metadata": {},
 "outputs": [],
 "source": []
},
{
 "cell_type": "markdown",
 "id": "cf2aff92",
 "metadata": {},
 "source": [
  "\n",
  "4. **[Evaluation]**:  \n",
  "   - Compare correlation of `Points` prediction vs actual `Points`.  \n",
  "   - Compare correlation of `cash_win` prediction vs actual `cash_win`.  \n",
  "   - Evaluate improvement over:  \n",
  "     - Raw `sum_projpts`.  \n",
  "     - Baseline rating (`0.5*projown + 0.5*projpts`).  \n"
 ]
```

NB_dkCashReview.ipynb (page 18)


```
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "20fa6fd1",
   "metadata": {},
   "outputs": [],
   "source": []
  },
  {
   "cell_type": "markdown",
   "id": "dc5a9516",
   "metadata": {},
   "source": [
    "\n",
    "5. **[Output]**:  \n",
    "   - Save final DataFrame with best `lineup_rating` to `final_entries.csv`.  \n",
    "   - Generate plots:  \n",
    "     - Feature importance for `cash_win` prediction.  \n",
    "     - Actual vs predicted `Points`.  \n",
    "\n",
    "**Coding Language**: Python (Pandas, Scikit-learn, NumPy).  \n",
    "**Goal**: Derive optimal lineup rating for cash-game success."
   ]
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "FantasySports_py3_12_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": "3.12.3"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```

utils\draftkingsCashOptimizer.py (page 1)

```
import pandas as pd
from pulp import LpMaximize, LpProblem, LpVariable, lpSum, LpStatus
import logging
from typing import List, Optional, Dict, Any


def column_str_standardize(df: pd.DataFrame) -> pd.DataFrame:
    """Convert all column names to lowercase and strip whitespace."""
    return df.rename(columns=lambda x: x.strip().lower())


def fetch_player_data(
    week: str="week01",
    dk_salaries_file: str = "DKSalaries.csv",
    dk_projections_file: str = "fantasy_footballers-nfl-dk-Main-projections.csv",
) -> pd.DataFrame:
    """
    Load and merge DraftKings salary and projection data.
    Args:
        dk_salaries_path: Path to DK salaries CSV.
        projections_path: Path to projections CSV.
    Returns:
        Merged DataFrame with standardized columns.
    """
    try:
        dk_df = column_str_standardize(pd.read_csv(f"data/{week}/{dk_salaries_file}"))
        proj_df = column_str_standardize(pd.read_csv(f"data/{week}/{dk_projections_file}"))
    except Exception as e:
        logging.error(f"Error loading data: {e}")
        raise
    try:
        merged = pd.merge(dk_df, proj_df, on=["id", "position"], how="inner")
        merged.drop(columns=["name_y"], inplace=True)
        merged.rename(columns={"name_x": "name"}, inplace=True)
        desired_columns = [
            "id",
            "name",
            "position",
            "roster position",
            "salary",
            "game info",
            "team",
            "avgpointspergame",
            "projpts",
            "projown",
        ]
        merged.to_csv(f"data/{week}/dk_player_data.csv", index=False)
        return merged[desired_columns]
    except Exception as e:
        logging.error(f"Error merging data: {e}")
        raise
```

utils\draftkingsCashOptimizer.py (page 2)

```
def normalize_features(df, feature_columns):
    """
    Normalize feature columns to have mean=0 and std=1.

    Args:
        df: DataFrame with features
        feature_columns: List of columns to normalize

    Returns:
        DataFrame: df with normalized features
    """
    df_normalized = df.copy()

    for col in feature_columns:
        if col in df.columns:
            mean = df[col].mean()
            std = df[col].std()
            if std > 0:  # Avoid division by zero
                df_normalized[col] = (df[col] - mean) / std

    return df_normalized

def generate_top_lineups(
    player_data: pd.DataFrame,
    num_lineups: int = 5,
    locked_players: Optional[List[Any]] = None,
    projown_wt: float = 0.1,
    projpts_wt: float = 0.9,
) -> List[Dict[str, Any]]:
    """
    Generate the top N optimal lineups, optionally locking specific players.
    Args:
        player_data: DataFrame of all available players.
        num_lineups: Number of unique lineups to generate.
        locked_players: List of player 'id' values to lock (optional).
    Returns:
        List of dicts with 'score' and 'lineup' DataFrame.
    """
    logger = logging.getLogger("LineupOptimizer")
    all_lineups = []
    found_player_ids = []

    # Validate locked players
    if locked_players:
        logger.info("--- Validating Locked Players ---")
        all_player_ids = set(player_data["id"])
        for pid in locked_players:
            if pid not in all_player_ids:
                raise ValueError(
                    f"Locked player with ID '{pid}' not found in player data."
                )
        locked_df = player_data[player_data["id"].isin(locked_players)]
```

utils\draftkingsCashOptimizer.py (page 3)

```
    locked_pos_counts = locked_df["position"].value_counts()
    logger.info(f"Locked Player Counts by Position:\n{locked_pos_counts}")
    if locked_pos_counts.get("QB", 0) > 1:
        logger.warning("Locking >1 QB will result in an infeasible lineup.")
    if locked_pos_counts.get("RB", 0) > 3:
        logger.warning("Locking >3 RBs will result in an infeasible lineup.")
    if locked_pos_counts.get("WR", 0) > 4:
        logger.warning("Locking >4 WRs will result in an infeasible lineup.")
    if locked_pos_counts.get("TE", 0) > 1:
        logger.warning("Locking >1 TEs will result in an infeasible lineup.")
    if locked_pos_counts.get("DST", 0) > 1:
        logger.warning("Locking >1 DEF will result in an infeasible lineup.")

for i in range(num_lineups):
    logger.info(f"--- Finding Lineup #{i + 1} ---")
    model = LpProblem(f"Fantasy_Lineup_{i + 1}", LpMaximize)
    player_vars = {
        row["id"]: LpVariable(f"player_{row['id']}", cat="Binary")
        for _, row in player_data.iterrows()
    }

    model += (
        lpSum(
            player_vars[row["id"]] * (0.1 * row["projpts"] + 0.9 * row["projown"])
            for _, row in player_data.iterrows()
        ),
        "Maximize_Figure_of_Merit",
    )

    model += lpSum(player_vars.values()) == 9, "Total_9_Players"
    model += (
        lpSum(
            player_vars[row["id"]] * row["salary"]
            for _, row in player_data.iterrows()
        )
        <= 50000,
        "Salary_Cap",
    )

    positions = ["QB", "RB", "WR", "TE", "DST"]
    for pos in positions:
        pos_players = player_data[player_data["position"] == pos]
        pos_count = lpSum(
            player_vars[row["id"]] for _, row in pos_players.iterrows()
        )
        if pos == "QB":
            model += pos_count <= 1, f"Max_1_QB_{i}"
        elif pos == "RB":
            model += 2 <= pos_count <= 3, f"RB_Range_{i}"
        elif pos == "WR":
            model += 3 <= pos_count <= 4, f"WR_Range_{i}"
        elif pos == "TE":
```

utils\draftkingsCashOptimizer.py (page 4)

```
            model += pos_count <= 1, f"Max_1_TE_{i}"
        elif pos == "DST":
            model += pos_count == 1, f"Exactly_1_DEF_{i}"

    # Add constraints for locked players
    if locked_players:
        for player_id in locked_players:
            model += player_vars[player_id] == 1, f"Lock_Player_{player_id}"

    # Exclude previously found lineups
    for prev_ids in found_player_ids:
        model += (
            lpSum(player_vars[pid] for pid in prev_ids) <= 8,
            f"Exclude_Lineup_{found_player_ids.index(prev_ids)}",
        )

    status = model.solve()

    if LpStatus[status] == "Optimal" and model.objective is not None:
        score = model.objective.value()
        selected_ids = [
            pid for pid, var in player_vars.items() if var.varValue == 1
        ]
        found_player_ids.append(selected_ids)
        lineup_df = player_data[player_data["id"].isin(selected_ids)].copy()
        all_lineups.append({"score": score, "lineup": lineup_df})
        logger.info(f"Status: Optimal. Score: {score:.2f}\n")
    else:
        logger.warning(
            f"Could not find an optimal solution for lineup #{i + 1}. Stopping."
        )
        break

return all_lineups
```

utils\draftkingsCashReview.py (page 1)

```
import pandas as pd


def load_split_dk_results(week:str):
    """
    Load and split DraftKings contest results into lineups and player pool.

    Args:
        week (str): The week identifier for the data folder.
        file_name (str): The name of the CSV file containing contest standings.
    """

    # Load the CSV file
    file_path = f'data/{week}/contest-standings.csv'
    df = pd.read_csv(file_path)

    # Identify the index of 'Unnamed: 6' (or similar placeholder)
    # This acts as the split point
    split_col = 'Unnamed: 6'

    if split_col not in df.columns:
        raise ValueError(f"Column '{split_col}' not found. Please check the actual name.")

    split_idx = df.columns.get_loc(split_col)

    # Split the DataFrame
    entries_df = df.iloc[:, :split_idx]  # Everything before 'Unnamed: 6'
    playerpool_df = df.iloc[:, split_idx:]  # From 'Unnamed: 6' onward


    # Clean up playerpool: drop the NaN column if it's just placeholder
    # Rename the first column to something meaningful if needed
    playerpool_df = playerpool_df.dropna(axis=1, how='all')  # Remove all-NaN columns
    return playerpool_df, entries_df
def load_player_data(week:str, playerpool_df):
    """
    Load player data for the given week.

    Args:
        week (str): The week identifier for the data folder.
    """
    player_data_path = f'data/{week}/dk_player_data.csv'
    player_data = pd.read_csv(player_data_path)
    player_results_df = playerpool_df.merge(player_data, left_on='Player', right_on='name',
how='left')
    # Strip ALL leading-and-trailing whitespace (tabs, new-lines, spaces, etc.)
    player_results_df['Player'] = player_results_df['Player'].str.strip()
    return player_results_df

def parse_lineup_string(lineup_str):
    """
    Parse the lineup string to extract individual player names
    Example: "DST Ravens FLEX George Pickens QB Jared Goff RB Christian McCaffrey..."
```

```
    """
    if pd.isna(lineup_str):
        return []

    # Split the lineup string and look for player names
    # Positions are typically: DST, QB, RB, WR, TE, FLEX
    positions = ['DST', 'QB', 'RB', 'WR', 'TE', 'FLEX']

    players = []
    parts = lineup_str.split()

    i = 0
    while i < len(parts):
        if parts[i] in positions:
            # Found a position, next part(s) should be player name
            i += 1
            player_name_parts = []

            # Collect name parts until we hit another position or end
            while i < len(parts) and parts[i] not in positions:
                player_name_parts.append(parts[i])
                i += 1

            if player_name_parts:
                player_name = ' '.join(player_name_parts)
                players.append(player_name)
        else:
            i += 1

    return players

def calculate_player_score(player_name, player_results_df,
                           projown_wt=0.5, projpts_wt=0.5):
    """
    Return weighted score for one player.
    """
    # locate the player (case-insensitive)
    mask = player_results_df['Player'].str.lower() == player_name.lower()
    player_match = player_results_df[mask]

    if player_match.empty:
        # player not found — log & give zero contribution
        print(f"⚠  '{player_name}' not found in player_results")
        return 0.0

    # take the first matching row
    player_row = player_match.iloc[0]

    projown = pd.to_numeric(player_row.get('projown', 0), errors='coerce')
    projpts = pd.to_numeric(player_row.get('projpts', 0), errors='coerce')

    projown = 0 if pd.isna(projown) else projown
```

utils\draftkingsCashReview.py (page 3)

```
    projpts = 0 if pd.isna(projpts) else projpts

    return projown_wt * projown + projpts_wt * projpts

def calculate_lineup_rating(entries_df, player_results_df, projown_wt=0.5, projpts_wt=0.5):
    """
    Calculate lineup_rating for each entry
    """
    lineup_ratings = []

    for idx, row in entries_df.iterrows():
        lineup_str = row['Lineup']
        players_in_lineup = parse_lineup_string(lineup_str)

        total_rating = 0
        player_count = 0

        for player in players_in_lineup:
            player_score = calculate_player_score(
                player, player_results_df, projown_wt, projpts_wt
            )
            total_rating += player_score
            player_count += 1

        lineup_ratings.append({
            'EntryId': row['EntryId'],
            'EntryName': row['EntryName'],
            'Rank': row['Rank'],
            'Points': row['Points'],
            'lineup_rating': total_rating,
            'players_found': player_count
        })


    return pd.DataFrame(lineup_ratings)

def engineer_lineup_features(lineup_str, player_results_df):
    """
    Calculate all required features for a given lineup.

    Args:
        lineup_str: String representation of the lineup
        player_results_df: DataFrame containing player projections

    Returns:
        dict: Dictionary of calculated features
    """
    # Parse lineup into player list if not already parsed
    if isinstance(lineup_str, str):
        players = parse_lineup_string(lineup_str)
    else:
        players = lineup_str
```

```
    # Initialize features dictionary
    features = {
        'sum_projpts': 0,
        'sum_projown': 0,
        'avg_floor': 0,
        'num_value_players': 0,
        'num_chalk_players': 0,
        'qb_wr_stacks': 0,
        'num_stacks': 0
    }

    # Track player data for stack calculation
    player_data = []

    # Calculate per-player features
    for player_name in players:
        # Find player in results df (case insensitive)
        mask = player_results_df['Player'].str.lower() == player_name.lower()
        if mask.any():
            player_row = player_results_df.loc[mask].iloc[0]

            # Basic sums
            features['sum_projpts'] += float(player_row.get('projpts', 0))
            proj_own = float(player_row.get('projown', 0))
            features['sum_projown'] += proj_own

            # Value and chalk calculations
            if proj_own < 20:
                features['num_value_players'] += 1
            if proj_own > 60:
                features['num_chalk_players'] += 1

            # Floor calculation
            features['avg_floor'] += float(player_row.get('avgpointspergame', 0))

            # Store player data for stack calculation
            player_data.append({
                'name': player_name,
                'position': player_row.get('rosterposition', ''),
                'team': player_row.get('teamabbrev', '')
            })

    # Calculate average floor
    if players:
        features['avg_floor'] /= len(players)


def add_lineup_features(entries_df, player_results_df):
    """
    Add all lineup features to entries dataframe.
```

```
    Args:
        entries_df: DataFrame with lineup entries
        player_results_df: DataFrame with player projections

    Returns:
        DataFrame: entries_df with added features
    """
    # Apply feature engineering function to each row
    features_list = entries_df['Lineup'].apply(
        lambda x: engineer_lineup_features(x, player_results_df)
    ).tolist()

    # Convert list of dicts to DataFrame and join with entries_df
    features_df = pd.DataFrame(features_list, index=entries_df.index)

    # Join features with original dataframe
    return pd.concat([entries_df, features_df], axis=1)

def normalize_features(df, feature_columns):
    """
    Normalize feature columns to have mean=0 and std=1.

    Args:
        df: DataFrame with features
        feature_columns: List of columns to normalize

    Returns:
        DataFrame: df with normalized features
    """
    df_normalized = df.copy()

    for col in feature_columns:
        if col in df.columns:
            mean = df[col].mean()
            std = df[col].std()
            if std > 0:  # Avoid division by zero
                df_normalized[col] = (df[col] - mean) / std

    return df_normalized
def calculate_baseline_rating(df):
    """
    Calculate baseline rating as 0.5*projown + 0.5*projpts.

    Args:
        df: DataFrame with sum_projown and sum_projpts columns

    Returns:
        Series: Baseline ratings
    """
    # Normalize first to ensure equal weighting
    norm_projpts = (df['sum_projpts'] - df['sum_projpts'].mean()) / df['sum_projpts'].std()
    norm_projown = (df['sum_projown'] - df['sum_projown'].mean()) / df['sum_projown'].std()
```

utils\draftkingsCashReview.py (page 6)

```
    return 0.5 * norm_projown + 0.5 * norm_projpts

def export_training_data(entries_df, feature_columns, target_columns=None,
filename='training_data.csv'):
    """
    Export training data with features and targets.

    Args:
        entries_df: DataFrame with features and targets
        feature_columns: List of feature column names
        target_columns: List of target column names (default: None)
        filename: Output CSV filename (default: 'training_data.csv')

    Returns:
        DataFrame: The exported data
    """
    # Select columns to export
    cols_to_export = feature_columns.copy()
    if target_columns:
        cols_to_export.extend(target_columns)

    # Select only existing columns
    existing_cols = [col for col in cols_to_export if col in entries_df.columns]
    export_df = entries_df[existing_cols].copy()

    # Export to CSV
    export_df.to_csv(filename, index=False)

    return export_df
```

Folder Structure (Python Files)

```
├── NB_dkCashLineup.ipynb
├── NB_dkCashReview.ipynb
└── utils
    ├── draftkingsCashOptimizer.py
    └── draftkingsCashReview.py
```