NBA ML Prediction System - Quick Start Guide

Get up and running with the NBA ML prediction system in 5 minutes.

📋 Prerequisites

1. **Backend setup complete** - Database initialized and configured
2. **Python environment** - Virtual environment activated
3. **Data collected** - At least some historical player stats in database

🚀 Quick Start (3 Steps)

Step 1: Install ML Dependencies

```
cd /home/ubuntu/betting_backend
source venv/bin/activate
pip install -r requirements.txt
```

Step 2: Collect Data (if you haven't already)

```
# This collects schedule, teams, and player stats (takes 30-60 minutes)
python collect_data.py --with-stats
```

Step 3: Train Models

```
# Train all models (takes 5-15 minutes depending on data)
python models/nba/train_models.py
```

Or for quick testing, train just one prop:

```
python models/nba/train_models.py --test
```

🎯 Generate Your First Predictions

Check if you have games today:

```
python scripts/generate_nba_predictions.py --check-only
```

Generate predictions:

```
python scripts/generate_nba_predictions.py
```

That's it! Your predictions are now in the database.

📊 View Your Predictions

Via Database:

```
psql -h localhost -U betting_user -d betting_analysis

# Query today's predictions
SELECT
    p.name as player,
    proj.prop_type,
    proj.projected_value,
    proj.confidence
FROM projections proj
JOIN players p ON proj.player_id = p.id
WHERE DATE(proj.created_at) = CURRENT_DATE
ORDER BY proj.confidence DESC
LIMIT 20;
```

Via Python:

```
from database.db_manager import db_manager

predictions = db_manager.execute_query("""
    SELECT p.name, proj.prop_type, proj.projected_value, proj.confidence
    FROM projections proj
    JOIN players p ON proj.player_id = p.id
    WHERE DATE(proj.created_at) = CURRENT_DATE
    ORDER BY proj.confidence DESC
    LIMIT 10
""")

for pred in predictions:
    print(f"{pred['name']}: {pred['prop_type']} = {pred['projected_value']} "
          f"(confidence: {pred['confidence']}%)")

db_manager.close()
```

🧪 Test the System

Run comprehensive tests:

```
python scripts/test_system.py
```

See example usage:

```
python models/nba/example_usage.py
```

💎 Find Value Bets

```python
from models.nba.predict import NBAPredictor
from models.nba.value_finder import ValueFinder

predictor = NBAPredictor()
value_finder = ValueFinder()

# Get predictions
predictions = predictor.predict_today_games()

# Your betting lines (from odds API)
betting_lines = {
    player_id: {
        'points': 25.5,
        'rebounds': 7.5,
        'assists': 8.5
    }
}

# Find value
best_bets = value_finder.find_best_values(
    predictions,
    betting_lines,
    min_confidence=65,
    top_n=10
)

for bet in best_bets:
    print(f"{bet['player_name']}: {bet['prop_type']} {bet['bet_direction']}")
    print(f"  Edge: {bet['edge']:+.1f} | EV: {bet['ev_pct']:+.1f}%")
    print(f"  Recommendation: {bet['recommendation']}\n")
```

⏰ Automate Daily Predictions

Set up a cron job:

```
crontab -e

# Add this line (runs at 8 AM daily)
0 8 * * * cd /home/ubuntu/betting_backend && /home/ubuntu/betting_backend/venv/bin/py-
thon scripts/generate_nba_predictions.py >> logs/predictions_cron.log 2>&1
```

📁 File Structure

```
betting_backend/
├── models/nba/                          # ML models
│   ├── config.py                        # Configuration
│   ├── feature_engineering.py           # Feature extraction
│   ├── train_models.py                  # Training pipeline
│   ├── predict.py                       # Prediction engine
│   ├── value_finder.py                  # Value bet finder
│   ├── example_usage.py                 # Examples
│   ├── README.md                        # Full documentation
│   └── saved_models/                    # Trained models
│       ├── points_*.joblib
│       └── ...
└── scripts/
    ├── generate_nba_predictions.py      # Daily predictions
    └── test_system.py                   # System tests
```

🔧 Common Commands

```python
# Train models
python models/nba/train_models.py

# Train specific props
python models/nba/train_models.py --prop-types points rebounds assists

# Generate predictions for today
python scripts/generate_nba_predictions.py

# Generate predictions for specific date
python scripts/generate_nba_predictions.py --date 2024-10-25

# Retrain and predict
python scripts/generate_nba_predictions.py --retrain

# Test system
python scripts/test_system.py

# View examples
python models/nba/example_usage.py
```

🎓 What the Models Predict

The system generates predictions for 10 prop types:

1. **Points** - Total points scored
2. **Rebounds** - Total rebounds
3. **Assists** - Total assists
4. **3-Pointers Made** - Three-point shots made
5. **Steals** - Total steals
6. **Blocks** - Total blocks
7. **Turnovers** - Total turnovers
8. **Double-Double** - Probability of double-double

9. **Field Goals Made** - Total field goals made
10. **Free Throws Made** - Total free throws made

Each prediction includes:
- **Predicted Value** - The model's prediction
- **Confidence Score** - 0-100 confidence level
- **Prediction Range** - Low to high estimate
- **Model Breakdown** - Individual model predictions

📈 Understanding Confidence Scores

- **80-100%** - High confidence, strong data, models agree
- **60-79%** - Moderate confidence, good for betting
- **40-59%** - Low confidence, use with caution
- **0-39%** - Very low confidence, avoid betting

💡 Pro Tips

1. **More data = better predictions**: Collect historical data regularly
2. **Retrain weekly**: Models improve with fresh data
3. **High confidence only**: Focus on predictions with 70%+ confidence
4. **Compare to lines**: Always compare predictions to actual betting lines
5. **Track accuracy**: Monitor model performance over time
6. **Value > prediction**: A great prediction isn't valuable if the line is bad

🐛 Troubleshooting

"No models found"

```
python models/nba/train_models.py
```

"Insufficient data"

```
python collect_data.py --with-stats
```

"No predictions generated"

Check if there are games today:

```
psql -h localhost -U betting_user -d betting_analysis -c "SELECT * FROM games WHERE sport='NBA' AND date=CURRENT_DATE;"
```

Models perform poorly

- Collect more historical data
- Tune hyperparameters in `models/nba/config.py`
- Retrain models weekly

📚 Learn More

- **Full Documentation**: See `models/nba/README.md`
- **Backend Guide**: See `README.md`
- **Examples**: Run `python models/nba/example_usage.py`
- **Tests**: Run `python scripts/test_system.py`

✅ Verification Checklist

- [ ] ML dependencies installed (`pip list | grep scikit`)
- [ ] Data collected (100+ player game stats)
- [ ] Models trained (check `models/nba/saved_models/`)
- [ ] Predictions generated successfully
- [ ] Predictions saved to database
- [ ] Can query predictions from database
- [ ] Value finder works
- [ ] Tests pass (`python scripts/test_system.py`)

🚀 Next Steps

1. ✅ **You're ready!** Start generating daily predictions
2. 📊 **Connect to dashboard** - View predictions in your betting dashboard
3. 🤖 **Automate** - Set up cron jobs for daily predictions
4. 📈 **Track performance** - Monitor prediction accuracy
5. 💰 **Find value** - Use value finder to identify profitable bets

---

Need Help?
- Check `models/nba/README.md` for detailed documentation
- Run `python scripts/test_system.py` to diagnose issues
- Review logs in `logs/` directory

Remember: This is for educational purposes. Always bet responsibly!