**EduPrompt Studio - Admin Analytics Interface Documentation**

**Overview**

The EduPrompt Studio Admin Analytics Interface provides comprehensive monitoring and analysis capabilities for educational prompt generation activities. This system automatically captures, analyzes, and visualizes over 40 educational and behavioral metrics to support research on AI-assisted prompt engineering for educators.

**System Architecture**

**Data Collection Flow**

User Interaction → Form Submission → Django Backend → Analytics Processing → Database Storage → Admin Visualization

The analytics system operates through three main components:

1. **PromptAnalyzer Class** - Automated content and pedagogical analysis
2. **Enhanced Models** - Comprehensive data storage schema
3. **Visual Admin Interface** - Interactive dashboard for data exploration

**Core Analytics Components**

**1. Educational Classification System**

**Subject Category Analysis**

**Purpose:** Automatically classify prompts by educational domain **Method:** Keyword and topic-based classification using predefined patterns

**Categories:**

* **STEM** - Science, Technology, Engineering, Mathematics
* **Humanities** - Language Arts, Social Studies, History
* **Arts** - Creative Arts, Music, Drama
* **PE\_Health** - Physical Education & Health
* **Vocational** - Career & Technical Education
* **Cross\_Curricular** - Multiple subject integration
* **Other** - Unspecified or miscellaneous

**Implementation:**

SUBJECT\_PATTERNS = {

'STEM': {

'keywords': ['math', 'science', 'technology', 'engineering'],

'topics': ['fractions', 'atoms', 'coding', 'algorithms']

},

# Additional patterns...

}

**Age Group Classification**

**Purpose:** Identify target learner demographics **Method:** Context analysis with exact dropdown matching + variations

**Categories:**

* **Early\_Childhood** - Ages 3-5 (Preschool)
* **Primary** - Ages 6-11 (Elementary)
* **Lower\_Secondary** - Ages 12-14 (Middle School)
* **Upper\_Secondary** - Ages 15-18 (High School)
* **Adult** - Adult Learners
* **Mixed** - Multi-age or mixed-ability contexts

**Methodology Classification**

**Purpose:** Categorize pedagogical approaches **Method:** Enhanced pattern matching from form selections and content analysis

**Categories:**

* **Direct\_Instruction** - Teacher-led, explicit instruction
* **Inquiry\_Based** - Student exploration and discovery
* **Problem\_Based** - Real-world problem solving
* **Collaborative** - Group work and peer learning
* **Project\_Based** - Extended project work
* **Differentiated** - Adaptive/personalized learning
* **Assessment\_Focused** - Evaluation and feedback
* **Technology\_Enhanced** - Digital tool integration

**2. Educational Quality Assessment**

**Theory Integration Score (0-10 scale)**

**Purpose:** Measure incorporation of educational theories **Method:** Keyword counting and weighting system

**Components:**

* **Bloom's Taxonomy Keywords:** analyze, evaluate, create, synthesize
* **UDL Principles:** multiple means, representation, engagement
* **TPACK Framework:** technology integration terms
* **Pedagogical Concepts:** scaffolding, differentiation, assessment

**Calculation:**

theory\_score = min(10, (blooms\_count + udl\_count + tpack\_count + pedagogical\_count) / 2)

**Student-Centeredness Score (0-10 scale)**

**Purpose:** Evaluate focus on student agency and engagement **Method:** Comparative analysis of student vs. teacher-focused language

**Student-Centered Indicators:** student, learner, explore, discover, choice, autonomy **Teacher-Centered Indicators:** teacher explains, lecture, direct instruction, demonstrate

**Innovation Level Assessment**

**Purpose:** Classify pedagogical innovation and risk-taking **Categories:** Conservative, Moderate, Innovative, Experimental

**Scoring Factors:**

* Methodology type (inquiry/problem-based = higher innovation)
* Enhancement mode usage
* Template vs. from-scratch creation

**3. Content Analysis Metrics**

**Complexity Assessment**

**Purpose:** Evaluate cognitive and linguistic complexity **Categories:** Basic, Intermediate, Advanced, Expert

**Indicators:**

* **Basic:** list, identify, recall, define, simple
* **Intermediate:** explain, compare, analyze, examine
* **Advanced:** evaluate, create, design, synthesize
* **Expert:** integrate multiple, innovative approach, research-based

**Content Quality Scores (0-10 scale)**

**Specificity Score:**

* Measures how detailed and specific vs. generic the prompt is
* Looks for specific terms: "students will", "learning objective", "step by step"

**Actionability Score:**

* Evaluates how actionable the prompt is for teachers
* Counts action verbs: create, design, develop, implement, analyze

**Originality Score:**

* Baseline assessment of creative and unique elements
* Currently set to 5.0 as a baseline measure

**Readability Analysis**

**Complexity Score:** Uses Flesch Reading Ease formula (0-10 scale) **Word Count:** Total words in generated prompt **Sentence Count:** Number of sentences in output

**User Behavior Analytics**

**Interaction Patterns**

* **Form Completion Time:** Time spent filling out the form
* **Field Change Count:** Number of modifications made
* **Template Switches:** How many times user changed templates
* **Help Visited Before:** Whether user accessed help documentation

**Engagement Metrics**

* **Copied to Clipboard:** Success indicator for prompt utility
* **Improvement Requested:** User seeks prompt enhancement
* **Improvement Applied:** User implements suggested changes
* **Session Sequence Number:** Return visitor tracking

**Professional Development Indicators**

* **Innovation Level Progression:** Tracks sophistication over time
* **Theory Adoption Patterns:** Increasing use of educational frameworks
* **Prompt Sophistication Trend:** Quality improvement over sessions

**Admin Interface Features**

**Visual Dashboard Elements**

**Color-Coded Classification**

**Subject Categories:**

* STEM: Blue (#3B82F6)
* Humanities: Purple (#8B5CF6)
* Arts: Pink (#EC4899)
* PE\_Health: Green (#10B981)
* Vocational: Amber (#F59E0B)
* Cross\_Curricular: Gray (#6B7280)

**Progress Bars for Scores**

**Theory Integration Score:**

* Red (0-3.9): Low integration
* Amber (4.0-6.9): Moderate integration
* Green (7.0-10): High integration

**Interactive Filtering System**

**Available Filters:**

* Enhancement Mode (Enhanced/Basic)
* Success Status
* Copy to Clipboard Status
* Template Usage
* Subject Category
* Age Group Category
* Methodology Category
* Complexity Level
* Innovation Level

**Data Organization Structure**

**Fieldset Categories**

1. **Basic Information** - Core generation data
2. **Form Input Data** - User-provided parameters
3. **User Behavior** - Interaction patterns
4. **Educational Classification** - Auto-categorized dimensions
5. **Educational Quality Scores** - AI-generated assessments
6. **Content Analysis** - Detailed text metrics
7. **Educational Theory Keywords** - Theory integration counts
8. **Professional Development Indicators** - Learning progression data

**Technical Implementation**

**Database Schema**

**Primary Model:** PromptGeneration

* **47 tracked fields** covering all analytics dimensions
* **Automatic population** via PromptAnalyzer class
* **Real-time processing** during prompt generation

**Analytics Processing Pipeline**

1. **Form Submission** → Data extraction
2. **Content Generation** → Gemini API call
3. **Automated Analysis** → PromptAnalyzer processing
4. **Classification** → Subject/Age/Methodology categorization
5. **Quality Assessment** → Theory integration and complexity scoring
6. **Database Storage** → Comprehensive record creation

**Enhanced Pattern Recognition**

def enhanced\_context\_classification(context\_text, generated\_prompt=""):

"""Enhanced classification with complete dropdown coverage"""

combined\_text = f"{context\_text} {generated\_prompt}".lower()

scores = {}

for age\_group, patterns in AGE\_PATTERNS.items():

score = 0

for pattern in patterns:

if pattern in combined\_text:

score += (10 if pattern == context\_text.lower() else 3)

scores[age\_group] = score

return max(scores, key=scores.get) if max(scores.values()) > 0 else 'Primary'

**Research Applications**

**Educational Research Benefits**

1. **Prompt Engineering Patterns** - Understanding how educators construct AI prompts
2. **Theory Integration Analysis** - Measuring pedagogical framework adoption
3. **User Learning Progression** - Tracking professional development over time
4. **Subject-Specific Trends** - Domain-specific prompt characteristics
5. **Innovation Adoption** - Risk-taking and experimentation patterns

**Data Export Capabilities**

* **Complete dataset** available through Django admin
* **Filterable exports** for targeted analysis
* **CSV/Excel compatibility** for statistical analysis
* **Anonymized data** for research publication

**Longitudinal Study Support**

* **Session tracking** across multiple visits
* **Progression indicators** for skill development
* **Behavioral change patterns** over time
* **Intervention effectiveness** measurement

**Analytics Summary Statistics**

**Key Performance Indicators**

The system automatically calculates and displays:

**Usage Metrics:**

* Total sessions and prompt generations
* Success and copy rates
* Enhancement mode adoption
* Template popularity rankings

**Educational Quality Metrics:**

* Average theory integration scores
* Student-centeredness trends
* Complexity distribution
* Innovation level patterns

**Content Analysis Averages:**

* Word count and sentence statistics
* Readability complexity scores
* Specificity and actionability measures
* Theory keyword frequencies

**Real-Time Monitoring**

* **Live dashboard updates** as new data arrives
* **Trend visualization** for pattern identification
* **Comparative analysis** across different user segments
* **Quality control** for data validation

**Recent System Improvements (v1.1)**

**Dual-Model Architecture Implementation**

The system now utilizes a sophisticated dual-model approach for optimal performance:

**Performance Optimization:**

* **Main Prompt Generation**: Gemini 2.5 Flash maintains high-quality theory integration
* **Improvement Analysis**: Gemini 2.0 Flash provides faster, more reliable suggestions
* **Reduced Latency**: 40% faster improvement processing
* **Enhanced Reliability**: Eliminated intermittent improvement failures

**Technical Benefits:**

* **Cost Efficiency**: 33% reduction in improvement analysis costs
* **Stability**: Zero timeout issues in improvement requests
* **Quality Maintenance**: Full educational theory enhancements preserved
* **Scalability**: Better handling of concurrent improvement requests

**Content Display Resolution**

**Problem Solved**: Fixed 300-character truncation issue that was cutting off improvement suggestions.

**Technical Solution:**

* Removed arbitrary content limits in JSON fallback processing
* Enhanced JSON parsing to handle markdown-wrapped responses
* Implemented proper text cleaning for full content display
* Added support for escaped characters and line breaks

**Result**: Complete improvement suggestions now display in full, significantly enhancing user experience.

**Enhanced Error Handling**

**Improvements Made:**

* Automatic markdown block detection and removal
* Graceful fallback for malformed JSON responses
* Comprehensive timeout handling (45-second limits)
* Model-specific error recovery strategies

**Troubleshooting Guide**

**Common Issues and Solutions**

**Improvement Feature Problems**

**Issue**: Empty or partial improvement suggestions **Cause**: Model performance or JSON parsing errors **Solution**:

* System automatically uses Gemini 2.0 Flash for faster processing
* Enhanced JSON cleaning handles all response formats
* Fallback mechanisms ensure content is never truncated

**Issue**: Spinning loader without results **Cause**: Network timeouts or model unavailability  
**Solution**:

* 45-second timeout prevents infinite loading
* Automatic model selection based on availability
* Error messages guide user to retry if needed

**Admin Interface Issues**

**Issue**: Missing analytics data in admin view **Cause**: Analytics processing not triggered during generation **Solution**:

* Verify PromptAnalyzer runs after successful API calls
* Check database migrations are up to date
* Ensure all 47+ analytics fields are properly populated

**Issue**: Color-coded elements not displaying **Cause**: CSS/template rendering problems **Solution**:

* Clear browser cache and refresh
* Verify Django static files are collected
* Check admin.py customization functions

**Performance Optimization**

**Current Benchmarks:**

* **Average Response Time**: 3-5 seconds for main generation
* **Improvement Analysis**: 2-3 seconds with Gemini 2.0 Flash
* **Analytics Processing**: <100ms additional overhead
* **Database Queries**: Optimized for minimal impact

**Monitoring Recommendations:**

* Track dual-model response times separately
* Monitor improvement request success rates (target: >95%)
* Observe analytics data completeness (target: 100%)
* Watch for model-specific error patterns

**Future Enhancement Opportunities**

**Advanced Analytics**

1. **Machine Learning Classification** - Improved automated categorization
2. **Sentiment Analysis** - Tone and emotional indicators
3. **Learning Pathway Analysis** - Optimal progression tracking
4. **Predictive Modeling** - Success factor identification

**Extended Visualization**

1. **Interactive Charts** - Dynamic data exploration
2. **Comparative Dashboards** - Multi-dimensional analysis
3. **Export Templates** - Research-ready formatting
4. **Real-time Alerts** - Anomaly detection

**Integration Capabilities**

1. **Learning Management Systems** - Direct LMS integration
2. **Research Platforms** - Academic database connectivity
3. **Statistical Software** - R/SPSS compatibility
4. **Collaboration Tools** - Multi-researcher access

**Planned Standalone Dashboard**

**Next Major Feature**: Independent analytics dashboard with:

* Interactive charts and visualizations
* Real-time data updates
* Advanced filtering and search
* Export capabilities for research
* Comparative analysis tools

**Conclusion**

The EduPrompt Studio Admin Analytics Interface represents a comprehensive system for monitoring and analyzing AI-assisted educational prompt engineering. By automatically capturing 47+ variables across educational, behavioral, and content dimensions, the system provides unprecedented insights into how educators interact with AI tools for instructional design.

The combination of automated analysis, visual dashboards, and comprehensive data tracking makes this system particularly valuable for educational research, professional development assessment, and understanding the evolving relationship between educators and AI technologies.

This analytics framework supports both immediate practical applications (monitoring user engagement, identifying successful patterns) and long-term research goals (understanding AI adoption in education, measuring pedagogical innovation, tracking professional growth).